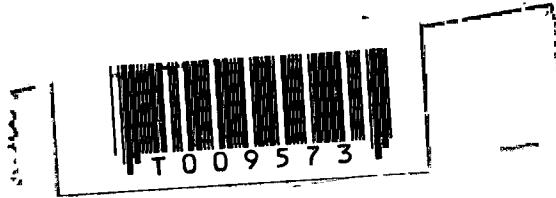




เครื่องอ่านบาร์โคดสำหรับใช้ในงานห้องสมุด

Bar Code Reader for Library Application



ปริญญา สุนวรางกูร

PRINYA SUMONVARANGKUL



อาจารย์ที่ปรึกษา  
รศ.ดร.สิทธิชัย ไกโคญอุดม  
ADVISOR  
ASSOC. PROF. DR. SITTHICHAI POOKAIYAUDOM

วิทยานิพนธ์สำหรับปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2529

เลขหมู่	
เลขทะเบียน	9573
วัน, เดือน, ปี	2.ค.ค. 2530

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่น  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อ	III
ABSTRACT	IV
บทที่ 1 บทนำ	-1-
บทที่ 2 ลักษณะโครงสร้างของบาร์โคด	-3-
2.1 รูปแบบโดยทั่วไปของบาร์โคด	-3-
2.2 หลักการออกแบบบาร์โคดสำหรับใช้ในระบบงานห้องสมุด	-5-
2.3 รายละเอียดเกี่ยวกับ Character Bar	-6-
2.4 การสร้างบาร์โคดสำหรับใช้เป็นรหัส	-7-
บทที่ 3 เครื่องอ่านบาร์โคด	-10-
3.1 อุปกรณ์สำหรับทำหน้าที่อ่านบาร์โคด	-10-
3.2 ระบบไมโครโปรเซสเซอร์ที่ออกแบบขึ้นเพื่อใช้เป็น เครื่องอ่านบาร์โคด	-13-
3.3 รายละเอียดเกี่ยวกับอุปกรณ์และวงจรที่ใช้ในระบบไมโคร- โปรเซสเซอร์	-14-
บทที่ 4 การสร้างโปรแกรมสำหรับเครื่องอ่านบาร์โคด	-23-
4.1 การกำหนดมาตรฐานในการอ่านบาร์โคด	-23-
4.2 ขั้นตอนการทำงานของโปรแกรมควบคุม	-24-
4.3 รายละเอียดเกี่ยวกับการทำงานของโปรแกรมต่าง ๆ ในระบบ	-25-
บทที่ 5 การใช้ไมโครคอมพิวเตอร์ในการพัฒนาระบบห้องสมุด	-32-
5.1 ระบบงานบริการทั่วไปในห้องสมุด	-32-
5.2 ระบบไมโครคอมพิวเตอร์	-33-

บทที่ 6	การออกแบบและพัฒนาโปรแกรมห้องสมุด	-38-
6.1	การออกแบบฐานข้อมูลสำหรับโปรแกรมห้องสมุด	-38-
6.2	การสร้างฐานข้อมูล	-39-
6.3	การสร้าง Index File ของแต่ละฐานข้อมูล	-40-
6.4	การออกแบบขั้นตอนการทำงานของโปรแกรม	-41-
6.5	การสร้างโปรแกรมห้องสมุด	-42-
6.6	การสร้างและพัฒนาโปรแกรมแอสเซมบลีเพื่อใช้ในการอ่านข้อมูล	-66-
6.7	รายละเอียดเกี่ยวกับโปรแกรมที่ใช้ในระบบห้องสมุด	-45-
บทที่ 7	บทสรุป	-47-
กิจกรรมประกาศ		-50-
เอกสารอ้างอิง		-51-
ภาคผนวก		
ภาคผนวกที่ 1	แสดงโปรแกรมของระบบไมโครโปรเซสเซอร์ ที่ใช้ในเครื่องอ่านบาร์โค้ด	-52-
ภาคผนวกที่ 2	แสดง File Layout ของฐานข้อมูลที่ใช้ใน โปรแกรมห้องสมุด	-60-
ภาคผนวกที่ 3	แสดง Screen Layout ของโปรแกรมห้องสมุด	-63-
ภาคผนวกที่ 4	แสดง Program Specification ของโปร แกรมห้องสมุด	-73-
ภาคผนวกที่ 5	แสดงโปรแกรมสำหรับใช้ในระบบห้องสมุด	-83-
ภาคผนวกที่ 6	วิธีใช้เครื่องอ่านบาร์โค้ด และไมโครคอมพิวเตอร์ ในงานห้องสมุด	-118-
ภาคผนวกที่ 7	รายละเอียดอุปกรณ์ TIL 180	-120-

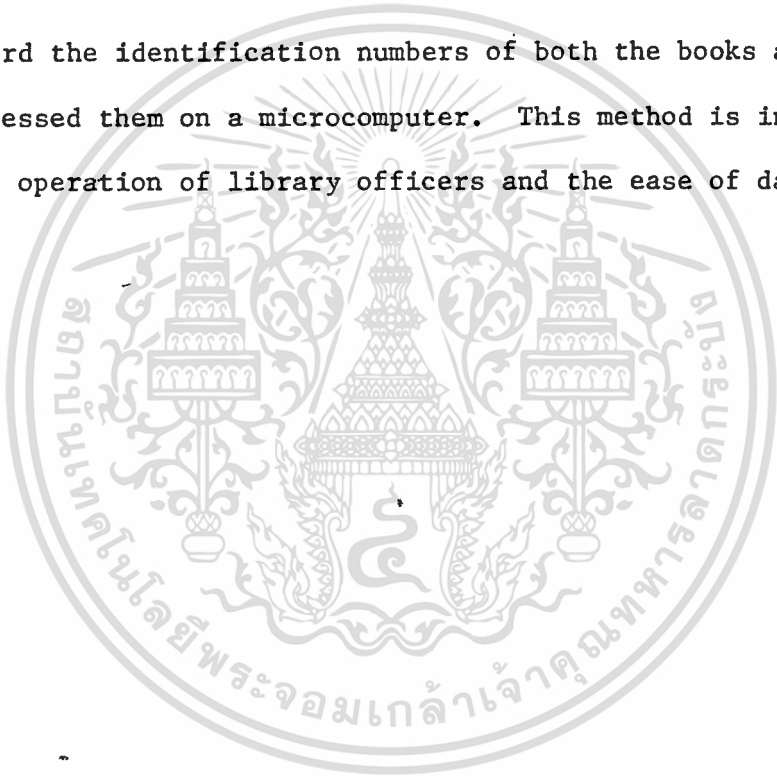
บทคัดย่อ

วิทยานิพนธ์นี้เสนอการประยุกต์นำเอา "บาร์โค้ด" (Bar Code) มาใช้ในการพัฒนาระบบบริการห้องสมุด ด้วยการนำเอาบาร์โค้ดมาใช้เป็น รหัสของหนังสือและผู้ใช้บริการ โดยใช้เครื่องอ่านบาร์โค้ด ร่วมกับไมโครคอมพิวเตอร์ในการทำงานในห้องสมุด ทั้งนี้ในระบบห้องสมุดแบบเดิมทั่วไปเจ้าหน้าที่ต้องทำการบันทึกรายละเอียดลงบนบัตรรายการต่าง ๆ ซึ่งอาจเกิดความล่าช้าผิดพลาด และอาจเกิดการสูญหายของบัตรบันทึกรายการได้เมื่อมีการใช้บริการเป็นจำนวนมากและยากแก่การตรวจสอบ ดังนั้นผลของการพัฒนาระบบบริการห้องสมุดนี้จึงเป็นการอำนวยความสะดวกให้แก่เจ้าหน้าที่ในการปฏิบัติงานและขจัดปัญหาดังกล่าว เพื่อเป็นการลดการสูญเสียที่อาจเกิดขึ้น



Abstract

This thesis presents an application of bar codes in the development of a library service system. Usually, service routines in library, i.e., borrowing and returning books, are performed by recording the details on documents which may be easily misplaced and difficult to check. To overcome these difficulties, a librarian can use the bar code reader to record the identification numbers of both the books and the users and processed them on a microcomputer. This method is intended to facilitate the operation of library officers and the ease of data modification.



บทที่ 1

บทนำ

ปัจจุบันความก้าวหน้าทางเทคโนโลยีในด้าน อิเล็กทรอนิกส์ และคอมพิวเตอร์ได้เจริญ รุดหน้าขึ้นอย่างรวดเร็ว และได้เข้าไปมีบทบาทในชีวิตประจำวันมากขึ้น โดยเฉพาะในด้านการบริการ ต่าง ๆ ซึ่งนับวันจะยิ่งขยายตัวเพิ่มขึ้นเรื่อย ๆ เนื่องจากมีความสะดวก รวดเร็ว และมีความแน่นอนสูง รวมไปถึงการประหยัดในด้านการใช้เอกสาร

ระบบงานของห้องสมุดมีลักษณะเป็นการบริการประเภทหนึ่งในด้านให้ความรู้ทั่วไปและ ข่าวสารต่าง ๆ ซึ่งย่อมที่จะต้องขยายตัวเพิ่มขึ้นตามวิวัฒนาการเจริญก้าวหน้าของความรู้ในทุกแขนง รวมทั้งการเพิ่มพูนของผู้ใช้บริการตามจำนวนประชากรที่เพิ่มขึ้นดังนั้นระบบการบริการห้องสมุดสมควรที่จะได้รับการพัฒนาให้ทันสมัยยิ่งขึ้น เพื่อรองรับการขยายตัวเพิ่มขึ้นของการบริการ

ระบบห้องสมุดแบบเดิมทั่วไป เจ้าหน้าที่ห้องสมุดต้องใช้เอกสารสำหรับจัดบันทึกข้อมูล ต่าง ๆ ลงในบัตรบันทึกรายการเมื่อมีการใช้บริการ ยืม และคืนหนังสือ และในการตรวจสอบข้อมูล ของหนังสือ เช่น หนังสือเล่มใดได้ถูกยืมไปบ้าง ใครเป็นผู้ที่ยืมหรือเลิกกำหนดส่งคืนหรือไม่ เหล่า นี้ก็ต้องตรวจสอบจากบัตรบันทึกรายการ ซึ่งอาจล่าช้าไม่สะดวก และอาจผิดพลาดหรือเกิดการสูญหาย ของบัตรได้ เมื่อมีการใช้บริการจำนวนมากด้วยเหตุผลเหล่านี้จึงได้เสนอการนำเอา การใช้บาร์- โทด (Bar Code) ร่วมกับระบบไมโครคอมพิวเตอร์มาใช้ในงานห้องสมุด เพื่อแก้ไขปัญหาดังกล่าว และอำนวยความสะดวกให้กับเจ้าหน้าที่ในการป้อนข้อมูล โดยเจ้าหน้าที่จะกระทำผ่าน เครื่องอ่านบาร์โคด (Bar Code Reader) ซึ่งควบคุมด้วยระบบไมโครโปรเซสเซอร์ ป้อนข้อมูลให้กับไมโครคอม- พิวเตอร์ทางพอร์ทอินพุทแบบอนุกรม (Serial Input Port RS-232 C) เพื่อความสะดวกในการ ที่จะนำไปใช้กับระบบไมโครคอมพิวเตอร์ทั่วไปในส่วนของโปรแกรมห้องสมุดสำหรับไมโครคอมพิวเตอร์ ได้สร้างขึ้นโดยใช้โปรแกรมสำเร็จรูปdBase-III PLUS ซึ่งมีความเหมาะสมในด้านการจัดการเกี่ยว กับฐานข้อมูล (Database Management) และสามารถเชื่อมโยงกับโปรแกรมที่อยู่ในรูปของ Binary File ในการรับข้อมูลจากเครื่องอ่านบาร์โคดเข้ามาทางพอร์ทอินพุทแบบอนุกรม (Serial Input Port) RS-232 C

เนื้อเรื่องในวิทยานิพนธ์นี้ได้จำแนกหัวข้อไว้เป็นบทต่าง ๆ ดังนี้

- บทที่ 1 กล่าวถึงความเป็นมาของเครื่องอ่านบาร์โคด และแนวความคิดของวิทยานิพนธ์
- บทที่ 2 เป็นการกล่าวถึงรายละเอียดเกี่ยวกับรูปแบบโครงสร้างและ การออกแบบของบาร์โคด ที่นำมาใช้ในระบบห้องสมุด
- บทที่ 3 เป็นการกล่าวถึงเครื่องอ่านบาร์โคดและ ระบบไมโครโปรเซสเซอร์ที่ใช้สำหรับควบคุม การทำงาน
- บทที่ 4 กล่าวถึงหลักการออกแบบและขั้นตอนการทำงานของโปรแกรมสำหรับ เครื่องอ่านบาร์โคด ในการอ่านโคดและส่งข้อมูลให้กับ เครื่องไมโครคอมพิวเตอร์
- บทที่ 5 กล่าวถึงการนำเอาเครื่องไมโครคอมพิวเตอร์มาพัฒนาใช้ในระบบห้องสมุดในการทำโปรแกรม ร่วมกับการใช้เครื่องอ่านบาร์โคด
- บทที่ 6 กล่าวถึงการพัฒนาและออกแบบ โปรแกรมห้องสมุดบน ไมโครคอมพิวเตอร์สำหรับการทำงาน ร่วมกับ เครื่องอ่านบาร์โคด
- บทที่ 7 เป็นการสรุปผลการทดลองใช้งาน และวิจารณ์ เพื่อเป็นแนวทางในการพัฒนาต่อไป

ภาคผนวกแสดงโปรแกรมที่ใช้ใน เครื่องอ่านบาร์โคดซึ่งควบคุมด้วยระบบไมโครโปร-เซสเซอร์โดยมี ไอซีเบอร์ Z-80 เป็นซีพียู และโปรแกรมห้องสมุดสำหรับไมโครคอมพิวเตอร์ ซึ่งประกอบด้วยรายละเอียดเกี่ยวกับฐานข้อมูลที่ใช้ การแสดงจอภาพ การออกแบบคุณสมบัติของโปรแกรม และโปรแกรมที่ใช้งานทั้งหมด รวมทั้งวิธีใช้เครื่องอ่านบาร์โคดและไมโครคอมพิวเตอร์ในการทำโปรแกรมห้องสมุด

## บทที่ 2

### ลักษณะโครงสร้างของบาร์โค้ด

บาร์โค้ด เป็นรหัสประเภทหนึ่งซึ่งประกอบด้วยแถบเส้นสีดำสลับกับแถบสีขาวเรียงขนานกันจำนวนหนึ่ง ซึ่งแถบเส้นเหล่านี้จะใช้เป็นรหัสแทนตัวเลขแบบทั่ว ๆ ไป โดยปกติบาร์โค้ดจะถูกนำไปใช้กับผลิตภัณฑ์สินค้าจำหน่ายปลีก (Retail Grocery Product) เพื่อบอกเกี่ยวกับเลขทะเบียนผู้ผลิตจะพิมพ์บาร์โค้ดไว้บนตัวหีบห่อของสินค้าและทางผู้จำหน่ายปลีก จะใช้อุปกรณ์สำหรับอ่านรหัสลากผ่านไปบนบาร์โค้ดเพื่อทำการคิดราคาสินค้า และจัดการเกี่ยวกับสต็อกสินค้า สำหรับบาร์โค้ดที่นำมาใช้ในวิทยานิพนธ์นี้ได้ออกแบบขึ้นตามความเหมาะสมของการทำงานในระบบงานห้องสมุด โดยในบทนี้จะกล่าวถึงรายละเอียดเกี่ยวกับบาร์โค้ดและการสร้างขึ้นเพื่อใช้งาน

#### 2.1 รูปแบบโดยทั่วไปของบาร์โค้ด

บาร์โค้ดที่ปรากฏอยู่ตามผลิตภัณฑ์สินค้าโดยทั่วไป จะมีรูปแบบตามมาตรฐานของ Universal Product Code (UPC) ซึ่งได้รับการออกแบบให้ง่ายต่อการใช้อุปกรณ์ในการอ่าน การพิมพ์บาร์โค้ด และการตรวจสอบ บาร์โค้ดในระบบของ UPC เป็นระบบแบบ 10 ตัว โดยข้อมูลตัวเลขทางซ้าย 5 ตัว จะบอกถึงหมายเลขทะเบียนผู้ผลิต และทางขวา 5 ตัว จะบอกถึงประเภทของสินค้า ดังแสดงในรูปที่ 2.1 สำหรับในบางผลิตภัณฑ์หรือบางประเทศอาจมีรูปแบบที่เปลี่ยนแปลงไปบ้าง เช่น เป็นระบบ 12 ตัว เลข หรือ 8 ตัว เลข เป็นต้น ซึ่งขึ้นอยู่กับความต้องการและความเหมาะสมที่แตกต่างกันไป

สำหรับบาร์โค้ดที่ได้นำมาใช้ในระบบงานห้องสมุดนี้เป็นระบบแบบ 6 ตัว เลขทั้งนี้ได้คัดแปลงเพื่อความเหมาะสมสำหรับการใช้งาน เนื่องจากได้นำเอาบาร์โค้ดมาใช้เป็นรหัสแทนเลขประจำตัวนักศึกษา (User I.D. Number) และเลขทะเบียนหนังสือ (Accession Number) ซึ่งจำนวนตัวเลขของทั้งสองประเภทประกอบด้วยจำนวน 6 ตัว เลขเหมือนกัน ลักษณะของบาร์โค้ดที่ออกแบบขึ้นใช้งานได้แสดงไว้ในรูปที่ 2.2 โดยหลักการการออกแบบ ส่วนใหญ่ยังคงใช้กฎเกณฑ์



รูปที่ ๒.๑ ลักษณะบาร์โคดของ UPC

ตามมาตรฐานของ UPC ทั้งนี้เพื่อความสะดวกและเป็นรูปแบบที่คืออยู่แล้ว โดยบาร์โคดแต่ละชุดจะประกอบด้วยส่วนที่ใช้แทนรหัสตัวเลข และส่วนที่ใช้เป็น "การ์ดบาร์" (Guard Bar) ซึ่งจะอยู่ทางด้านซ้าย (Left Guard Bar) และทางด้านขวา (Right Guard Bar) ของบาร์ที่ใช้แทนตัวเลข สำหรับการ์ดบาร์นี้จะถูกใช้สำหรับตรวจสอบการเริ่มต้นและสิ้นสุดของการอ่านด้วยระบบซอฟต์แวร์ (Software) โดยการ์ดบาร์นี้จะไม่ใช่เป็นข้อมูลสำหรับการอ่านโคด



รูปที่ 2.2 ลักษณะของบาร์โคดที่ออกแบบขึ้นใช้งาน

## 2.2 หลักการออกแบบบาร์โคดสำหรับใช้ในระบบงานห้องสมุด

ลักษณะโครงสร้างทั่วไปของบาร์โคด จะประกอบด้วยบาร์สีดำและบาร์สีขาว เรียงขนานสลับกัน เป็นแถบรหัส มีขนาดความกว้างของบาร์แตกต่างกันไปตาม รหัสตัวเลขที่ใช้แทน สำหรับการออกแบบบาร์โคดในระบบแบบ 6 ตัวเลข มีข้อกำหนดในการออกแบบดังมี

- บาร์โคดแต่ละชุด ประกอบด้วยบาร์สีดำ (Black Bar) จำนวน 17 บาร์ และบาร์สีขาว (White Bar) จำนวน 16 บาร์
- ลักษณะของบาร์แต่ละอันมีลักษณะเป็น สีเหลี่ยมผืนผ้า
- บาร์ที่ใช้แทนตัวเลขจะเรียกว่า Character Bar โดยแต่ละตัวเลขที่ใช้แทนจะประกอบด้วย บาร์สีดำจำนวน 2 บาร์ และบาร์สีขาวจำนวน 2 บาร์
- ใน 1 Character จะถูกแบ่งเป็นส่วนย่อยอีก 7 ส่วนเท่าๆ กันซึ่งแต่ละส่วนที่ถูกแบ่งจะเรียกว่า "โมดูล" (Module) ดังแสดงในรูปที่ 2.3

- ในชุดแต่ละอันของ Character อาจเป็นสีดำ หรือสีขาวขึ้นอยู่กับแต่ละรหัสตัวเลขที่ใช้แทน
- ในแต่ละ Character จะเริ่มด้วยบาร์สีขาว และสิ้นสุดด้วยบาร์สีดำ เสมอ
- บาร์โคดแต่ละชุดจะต้องมีการคบาร์ จำนวน 2 ชุด อยู่ทางด้านซ้ายและด้านขวาของบาร์ตัวเลข
- จะต้องแสดงตัวเลขกำกับแต่ละ Character Bar เพื่อให้ทราบถึงว่าเป็น Character Bar ของเลขอะไร เพื่อสามารถตรวจสอบได้



จากรูปแสดง Character ของตัวเลข "0" ซึ่งเขียนเป็นไบนารีได้เป็น 0001101

รูปที่ 2.3 แสดงโครงสร้างของ Character Bar

### 2.3 รายละเอียดเกี่ยวกับ Character Bar

สำหรับรูปแบบ Character Bar ที่ใช้แทนตัวเลข ได้ใช้รูปแบบตามมาตรฐานของ ซึ่งได้ถูกออกแบบไว้โดยคำนึงถึงความปลอดภัยเคลื่อนที่อาจเกิดจากการพิมพ์บาร์โคด ทั้งนี้เพื่อป้องกันมิให้เกิดการอ่านผิดตัวเลขจากการใช้อุปกรณ์ในการอ่าน Character Bar ที่ใช้แทนรหัส

ตัวเลข สามารถแสดงสัญลักษณ์ในรูปของเลขไบนารี (Binary) แทนได้ ดังตารางในรูปที่ 2.4 โดยกำหนดให้ 1'S เป็นสัญลักษณ์ใช้แทนโมดูลสีดำและ 0'S เป็นสัญลักษณ์ใช้แทนโมดูลสีขาว

ตัวเลขฐานสิบ	สัญลักษณ์ในรูปไบนารี
0	0001101
1	0011001
2	0010011
3	0111101
4	0100011
5	0110001
6	0101111
7	0111011
8	0110111
9	0001011

รูปที่ 2.4 แสดงสัญลักษณ์ในรูปของเลขไบนารี

สำหรับการคีย์ทางคานซ้ายของบาร์โคดสามารถเขียนในรูปของเลขไบนารีได้เป็น 101 และทางคานขวาเป็น 010101 ในส่วนของการคีย์นี้ ไม่ได้ถูกใช้สำหรับเป็นข้อมูล

#### 2.4 การสร้างบาร์โคดสำหรับใช้เป็นตัวส

ในการนำเอาบาร์โคดมาใช้เป็น รหัสในระบบห้องสมุดนี้ ขนาดของโคดควรมีขนาดเล็กเหมาะสมกับการนำไปคอบนหนังสือ และบัตรประจำตัวของผู้ใช้บริการ ดังนั้นเพื่อความสะดวกในการสร้างบาร์โคด จึงได้ออกแบบต้นแบบบาร์โคดโดยให้มีขนาดใหญ่กว่าขนาดที่ใช้งานจริง จาก

จากนั้นจึงนำคั่นแบบที่ได้ทำการย่อขนาดลงจนได้ขนาดตามที่ต้องการ คั่นแบบที่สร้างขึ้นเป็นคั่นแบบของ Character Bar สำหรับแต่ละตัวเลข ทั้งนี้เพื่อความสะดวกในการนำมาเรียงประกอบกันเป็นรหัสต่าง ๆ และคั่นแบบของบาร์โค้ด ดังรูปที่ 2.5 ซึ่งแสดงถึงขนาดเท่าของจริงของคั่นแบบบาร์โค้ดที่ได้สร้างขึ้น ซึ่งมีขนาดของโมดูลเท่ากับ 2 มิลลิเมตร



รูปที่ 2.5 แสดงขนาดของคั่นแบบบาร์โค้ด

จากคั่นแบบบาร์โค้ดที่ได้ จะถูกนำมาย่อขนาดให้เล็กลงโดยจนได้ขนาดที่ต้องการดังรูปที่ 2.6 ซึ่งเป็นขนาดของบาร์โค้ดที่ใช้งานจริงขนาดโมดูลของ UPC มีขนาดมาตรฐานเท่ากับ 0.013 นิ้ว ส่วนขนาดโมดูลของบาร์โค้ดที่ได้สร้างขึ้นมีขนาดประมาณเท่ากับ 0.0196 นิ้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงขนาดของบาร์โค้ดที่ใช้งานจริง

อันที่จริงขนาดของบาร์โค้ดไม่ได้กำหนดแน่นอนตายตัวว่าต้องมีขนาดเท่าใดเพียงแต่คำนึงถึงความเหมาะสมในการนำไปใช้งานเป็นสำคัญ ทั้งนี้ระบบซอฟต์แวร์สำหรับอ่านรหัสสามารถตรวจสอบได้ โดยใช้ความสัมพันธ์ระหว่างขนาดของบาร์ และ ความเร็วในการอ่านสำหรับนำไปใช้ในการประมวลผล

### บทที่ 3

#### เครื่องอ่านบาร์โค้ด

ในการนำเอาบาร์โค้ดมาใช้งานร่วมกับระบบไมโครคอมพิวเตอร์ เพื่อทำโปรแกรมห้องสมุด จำเป็นต้องมีอุปกรณ์สำหรับใช้ในการอ่านบาร์โค้ด เพื่อบันทึกข้อมูลให้กับไมโครคอมพิวเตอร์ อุปกรณ์นี้คือเครื่องอ่านบาร์โค้ด (Var Code Reader) <sup>b</sup> ซึ่งควบคุมการทำงานด้วยระบบไมโครโปรเซสเซอร์ โดยเครื่องอ่านบาร์โค้ดนี้จะประกอบด้วยอุปกรณ์ส่วนที่ใช้ในการอ่านบาร์โค้ดและระบบไมโครโปรเซสเซอร์ซึ่งทำหน้าที่ควบคุมการทำงานและตรวจสอบจากนั้นจึงส่งข้อมูลไปยังไมโครคอมพิวเตอร์ โดยทำการส่งแบบอนุกรม (Serial) RS-232C เพื่อนำไปประมวลผลต่อไป

#### 3.1 อุปกรณ์สำหรับทำหน้าที่อ่านบาร์โค้ด

อุปกรณ์สำหรับใช้ในการอ่านบาร์โค้ด สามารถแบ่งออกได้เป็น 2 ส่วน คือส่วนที่ใช้เป็นหัวอ่าน และส่วนอินเทอร์เฟสกับระบบไมโครโปรเซสเซอร์ซึ่งอยู่บนบอร์ด โดยส่วนที่ใช้เป็นหัวอ่านเป็นอุปกรณ์สำหรับเจ้าหน้าที่ใช้จับลากผ่านไปบนบาร์โค้ดในขณะที่ทำการอ่านและส่วนอินเทอร์เฟส จะทำการขยายสัญญาณจากหัวอ่าน เพื่อบันทึกสัญญาณให้กับระบบไมโครโปรเซสเซอร์ต่อไป สำหรับรายละเอียดการทำงานของอุปกรณ์มีดังนี้

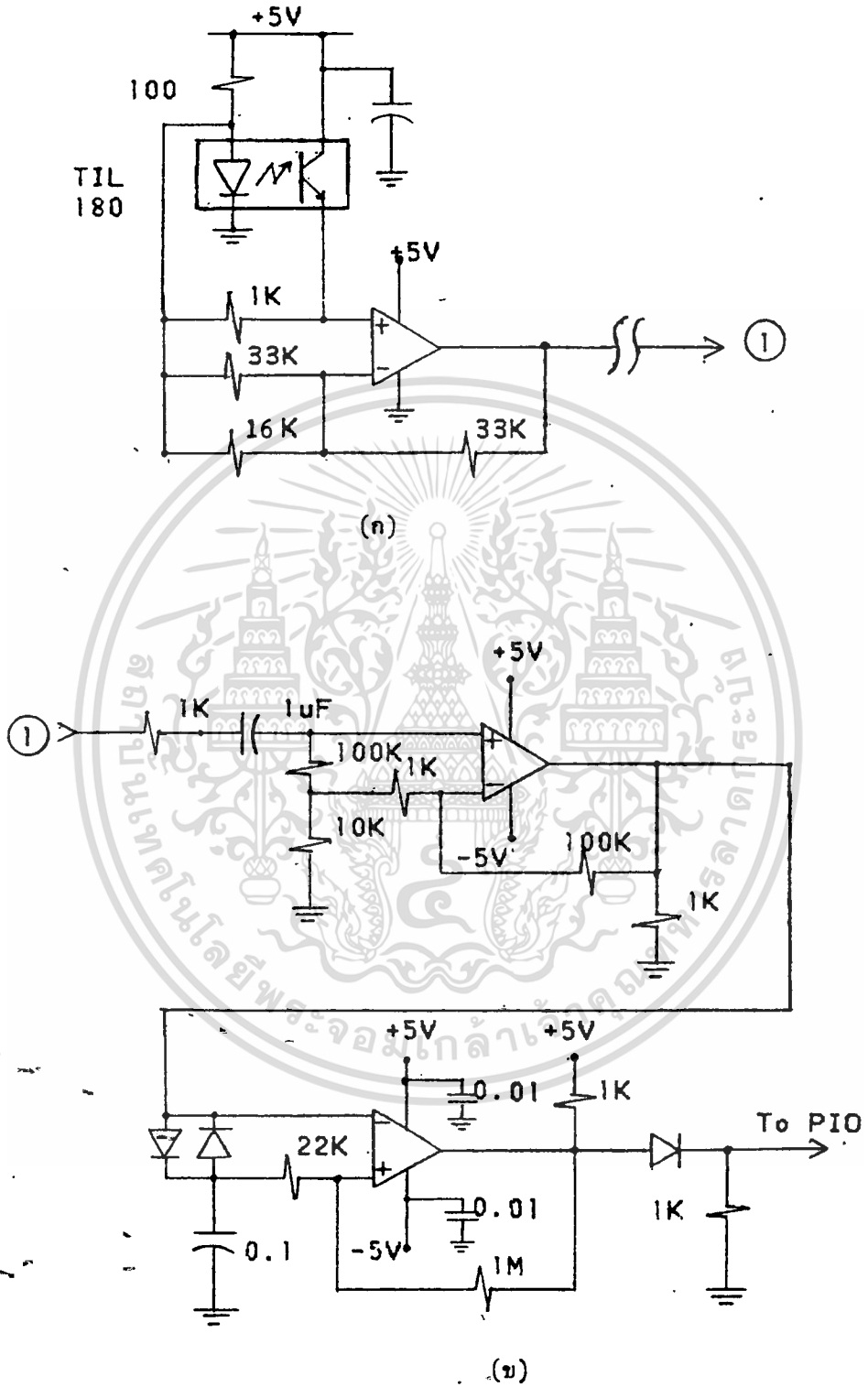
##### 3.1.1 หัวอ่านบาร์โค้ด (Wand)

ภายในของอุปกรณ์หัวอ่านนี้ประกอบด้วยอุปกรณ์ส่งและตรวจจับแสงอินฟราเรด (Infrared Sensor and Emitter) แบบ TIL 180 (รายละเอียดของ TIL 180 ดูได้ในภาคผนวก) โดยมีวงจรใช้งานดังแสดงในรูปที่ 3.1 (ก) ซึ่งภายในของ TIL 180 ประกอบด้วย ไดโอดอินฟราเรด (Infrared LED) ซึ่งทำหน้าที่เป็นตัวให้แสง (Emitter) อินฟราเรด และ Phototransistor ซึ่งทำหน้าที่เป็นตัวรับ (Sensor) โดยตัวรับจะทำงานก็คือเมื่อมีการสะท้อนกลับของแสงอินฟราเรดจากตัวส่งในขณะที่หัวอ่านผ่านไปบนบาร์โค้ด และเมื่อขณะผ่านไปบนบาร์โค้ด ค่า แสงอินฟราเรดจะถูกดูดซับไว้ (Absorb) ทำให้ตัวรับไม่ทำงาน จากวงจรในรูป 3.1 (ก)

ซึ่งเป็นวงจร Op-Amp. (Operation Amplifier) แบบ Noninverting Amp. ทำหน้าที่เป็น Buffer ซึ่งมีค่า Gain ประมาณ 3 เท่า ในขณะที่มีการสะท้อนของแสงอินฟราเรด (ขณะผ่านบาร์สีขาว) จะทำให้เกิดกระแสไหลผ่าน R 1K ทำให้ IC 1 ทำงาน เกิดสัญญาณที่เอาต์พุต (High) โดยสัญญาณเอาต์พุตที่ได้จะเปลี่ยนแปลงไปตามลักษณะของบาร์โค้ดที่หัวอ่านผ่านไป จากนั้นจึงป้อนสัญญาณให้กับส่วนอินเทอร์เฟสต่อไป

### 3.1.2 วงจรอินเทอร์เฟส

สัญญาณที่ได้จากหัวอ่านบาร์โค้ดจะถูกป้อนให้กับวงจรอินเทอร์เฟส เพื่อทำหน้าที่ขยายสัญญาณ และส่งไปยังระบบไมโครโปรเซสเซอร์ สำหรับในส่วนของวงจรอินเทอร์เฟสนี้ประกอบด้วยวงจร Op-Amp. 2 Stage โดยใน Stage แรกมี IC 2 ทำหน้าที่เป็น Noninverting Amp. มีค่า Gain ประมาณ 100 เท่าทำการขยายสัญญาณที่ได้จากหัวอ่าน และเอาต์พุตที่ได้จาก Stage นี้จะถูกนำไปผ่านวงจร Clipping เพื่อปรับรูปสัญญาณให้มีขนาดเป็น  $\pm 0.6$  Volt สำหรับป้อนให้กับ Stage สุดท้ายซึ่งมี IC 3 ทำงานแบบ Inverting Amp. ทำหน้าที่เป็นวงจร Switching โดยสัญญาณเอาต์พุตที่ได้จะ Saturate และกลับเฟสกับอินพุตสำหรับเอาต์พุตจาก Stage นี้ จะผ่านไดโอด เพื่อให้ได้สัญญาณเป็นแบบลอจิกทีทีแอล (TTL) โดยที่สัญญาณลอจิก "1" จะเป็นสถานะของการอ่านบาร์สีดำ และลอจิก "0" จะเป็นสถานะของการบาร์สีขาว

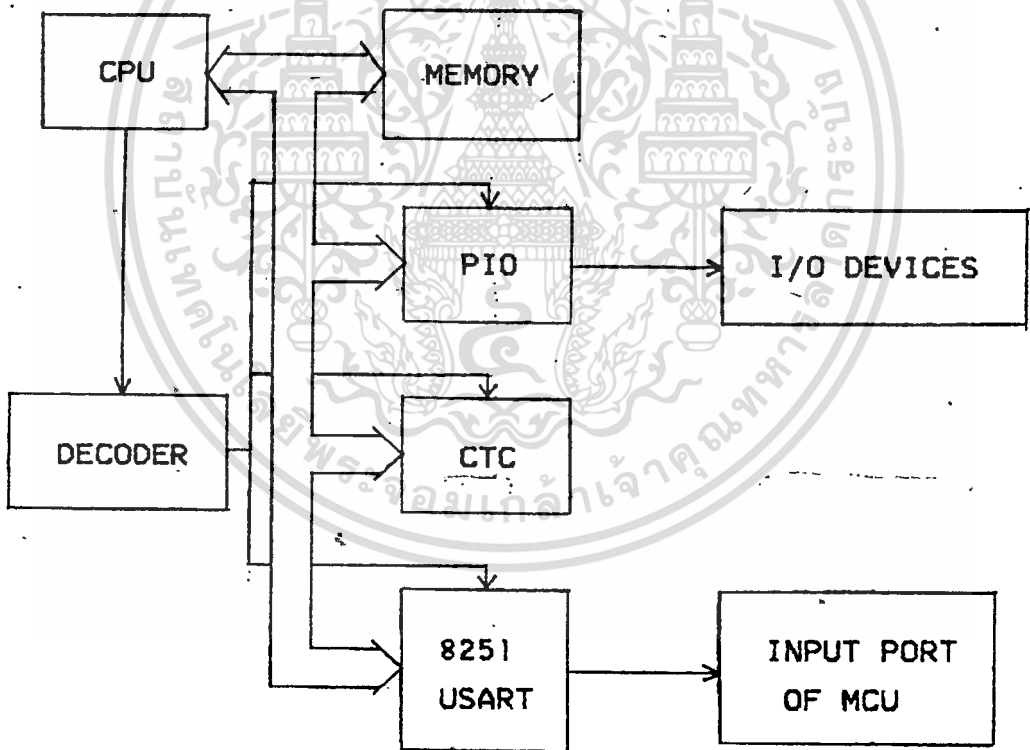


รูปที่ 3.1 (ก) แสดงวงจรใช้งานในส่วนที่ใช้เป็นหัวอ่าน  
(ข) แสดงวงจรส่วนอินเทอร์เฟซกับระบบไมโครโปรเซสเซอร์บนบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 ระบบไมโครโปรเซสเซอร์ที่ออกแบบขึ้นเพื่อใช้เป็นเครื่องอ่านบาร์โค้ด

ระบบไมโครโปรเซสเซอร์ที่ใช้เป็นเครื่องอ่านบาร์โค้ด มีหน้าที่ในการอ่านสัญญาณจากหัวอ่านบาร์โค้ด นำมาประมวลผล แล้วจึงส่งข้อมูลไปยังไมโครคอมพิวเตอร์แบบอนุกรม โดยระบบที่ออกแบบนี้ได้ใช้ Z-80 เป็น ซีพียู (CPU) และส่วนที่ซีพียูใช้ติดต่อกับคือ ส่วนหน่วยความจำ (memory) และส่วนที่ใช้ติดต่อกับอุปกรณ์ภายนอก (I/O Interface) ซึ่งประกอบด้วย Z-80 PIO Z-80 CTC และ 8251 USART เนื่องจากมีอุปกรณ์ที่ใช้อินเตอร์เฟสหลายตัว ดังนั้นจึงต้องมีวงจรถอดรหัส (Decoder) เพื่อให้ซีพียูเลือกใช้ตามต้องการ โดยโครงสร้างของระบบไมโครโปรเซสเซอร์ที่ใช้ สามารถเขียนเป็นบล็อกไดอะแกรม (Block Diagram) ได้ดังรูปที่ 3.2



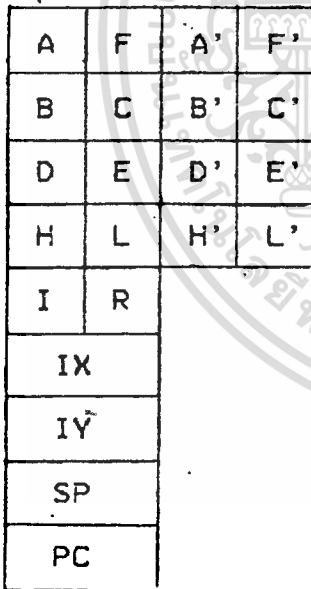
รูปที่ 3.2 แสดงบล็อกไดอะแกรมของระบบไมโครโปรเซสเซอร์ที่ใช้เป็นเครื่องอ่านบาร์โค้ด

### 3.3 รายละเอียดเกี่ยวกับอุปกรณ์และวงจรที่ใช้ในระบบไมโครโปรเซสเซอร์

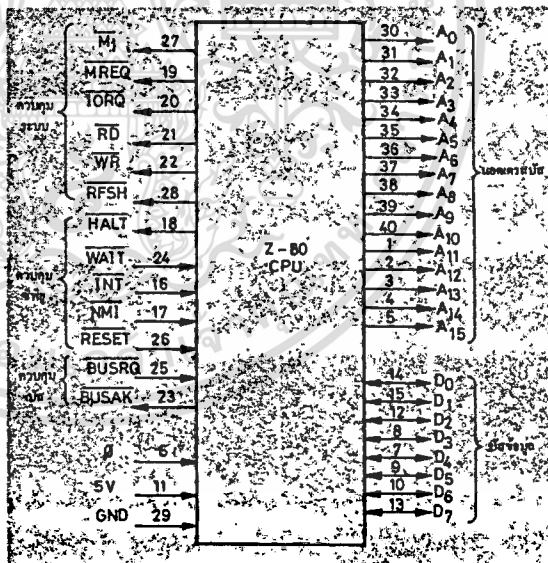
อุปกรณ์ต่าง ๆ ในระบบไมโครโปรเซสเซอร์ที่ใช้เป็นเครื่องอ่านบาร์โค้ดมีรายละเอียดเกี่ยวกับการทำงานดังนี้

#### 3.3.1 Z-80 CPU

ซีพียูใช้ Z-80 เป็นตัวควบคุมการทำงานทั้งหมด ซึ่งเป็นไมโครโปรเซสเซอร์ขนาด 8 บิต ( $D_0$  ถึง  $D_7$ ) มีแอดเดรสบัสขนาด 16 บิต ( $A_0$  ถึง  $A_{15}$ ) มีรีจิสเตอร์หลักขนาด 8 บิตสำหรับใช้งานทั่วไป 8 ตัว คือ A, F, B, C, D, E, H และ L มีรีจิสเตอร์สำรองขนาด 8 บิต ใช้งานอีก 8 ตัว คือ A', F', B', C', D', E', H', และ L' มีรีจิสเตอร์ใช้งานเฉพาะขนาด 16 บิต อีก 4 ตัวคือ IX, IY, SP และ PC และมีคำสั่งใช้งานได้ถึง 158 คำสั่งใช้งานได้ถึง 158 คำสั่งมีโครงสร้าง (Architecture) ดังแสดงในรูปที่ 3.3



(ก)



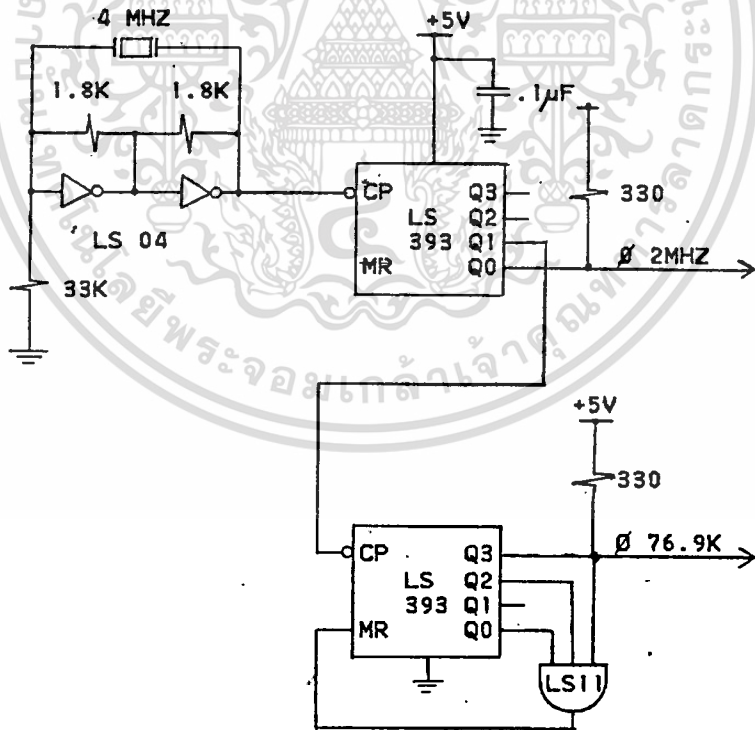
(ข)

รูปที่ 3.3 (ก) รีจิสเตอร์ภายใน Z-80 CPU

(ข) ขาสัญญาณต่าง ๆ ของ Z-80 CPU

### 3.3.2 ส่วนของสัญญาณนาฬิกา (Clock)

สัญญาณนาฬิกาที่ใช้ในระบบไมโครโปรเซสเซอร์ที่ได้จากวงจรรอสซิลเลเตอร์ ดังแสดงในรูปที่ 3.4 โดยใช้ไอซีเบอร์ 74 LS 04 ซึ่งเป็น Inverter และใช้คริสตอล (Crystal) ขนาด 4 MHz เป็นตัวกำเนิดความถี่ จากนั้นจึงนำสัญญาณที่ได้ม้อนให้กับ ไอซีเบอร์ 74 LS 393 ซึ่งเป็น Dual 4-Bit Binary Counter สัญญาณนาฬิกาที่ใช้ในระบบมีจำนวน 2 ชุด โดยชุดแรกได้จากการใช้ Binary Counter ตัวแรกเป็นวงจรรทวน 2 เพื่อให้ได้สัญญาณนาฬิกาขนาด 2 MHz ม้อนให้กับ CPU, PIO, CTC และ 8251 USART สำหรับสัญญาณนาฬิกาชุดที่สองได้จากการใช้ Binary Counter ตัวที่สองเป็นวงจรรทวน 13 โดยอาศัยสัญญาณจากขา Q<sub>1</sub> ของตัวแรก ซึ่งมีความถี่เท่ากับ 1 MHz เป็นอินพุต โดยสัญญาณนาฬิกาชุดที่สองนี้จะเท่ากับ 76.92 KHz หรือ ประมาณ  $4800 \times 16$  ซึ่งจะใช้เป็นความถี่ม้อนให้กับขา TXC ของ 8251 เพื่อให้ได้อัตราการส่ง (Baud Rate) เท่ากับ 4800 Baud



รูปที่ 3.4 แสดงรายละเอียดของวงจรสร้างสัญญาณของนาฬิกาที่ใช้ในระบบไมโครโปรเซสเซอร์

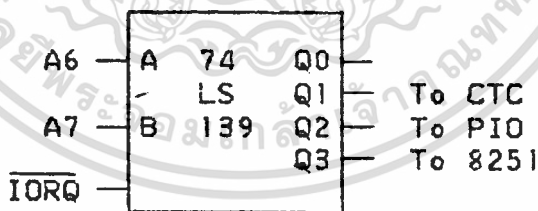
### 3.3.3 หน่วยความจำ (Memory)

ในส่วนของหน่วยความจำ สามารถแบ่งออกได้เป็น 2 ส่วนคือ หน่วยความจำชนิด ROM (Read Only Memory) โดยใช้อินทิเกรต 2716 ซึ่งเป็น EPROM ขนาด  $2k \times 8 \text{ bit}$  ซึ่งใช้เก็บ Monitor Program และหน่วยความจำชนิด RAM (Random Access Memory) โดยใช้อินทิเกรต 6116 ซึ่งเป็น Static RAM ขนาด  $2k \times 8 \text{ bit}$

ตำแหน่งแอดเดรสของ ROM จะเริ่มที่ตำแหน่ง 0000H ถึง 07FFH และแอดเดรสของ RAM จะเริ่มที่ตำแหน่ง 1000H ถึง 17FFH

### 3.3.4 ส่วนอุปกรณ์ที่ใช้ติดต่อกับอินพุตและเอาต์พุต (I/O)

อุปกรณ์ที่นำมา อินเทอร์เฟสกับซีพียู ซึ่งทำหน้าที่เป็นตัวกลางสำหรับ ซีพียู ใช้ติดต่อกับอุปกรณ์ อินพุต และเอาต์พุต ประกอบด้วย Z-80 PIO, Z-80 CTC และ 8251 USART โดยซีพียูจะเลือกติดต่อกับอุปกรณ์ตัวใด ก็ด้วยการเลือกผ่านวงจรถอดรหัส (Decoder) โดยใช้อินทิเกรตเบอร์ 74 LS 139 (2 to 4 Lines Decoder) และการ Decode ของซีพียูก็ โดยการใช้แอดเดรสบิต  $A_6$  และ  $A_7$  เป็นตัวกำหนด ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 แสดงวงจรถอดรหัส



การเลือกพอร์ทของอุปกรณ์สามารถแสดงได้ดังแสดงตารางรูปที่ 3.6

A <sub>7</sub>	A <sub>6</sub>	อุปกรณ์ที่ ติดต่อ
0	0	ไม่ใช่
0	1	CTC
1	0	PIO
1	1	8251

รูปที่ 3.6 แสดงตารางการเลือกพอร์ท

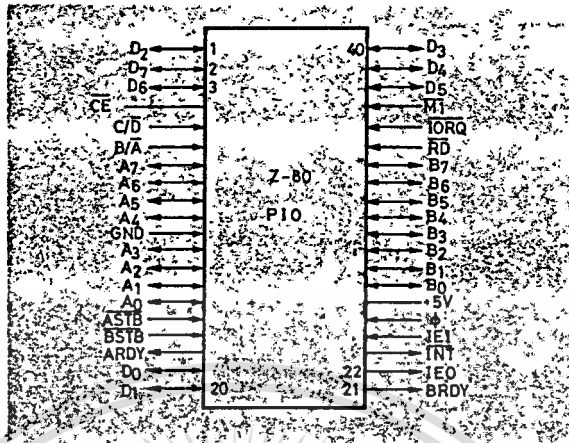
สำหรับรายละเอียดโดยสังเขปของอุปกรณ์ที่นำมาอินเทอร์เฟสกับซีพียูมีดังต่อไปนี้

Z-80 PIO (Parallel I/O)

เป็นอุปกรณ์ที่ใช้ในการอินเทอร์เฟสกับอุปกรณ์ อินพุต-เอาต์พุต แบบขนาน โดยมีหน้าที่เป็นตัวผ่านข้อมูลระหว่าง ซีพียู กับอุปกรณ์ I/O โดยผ่านทางบัสของระบบและพอร์ท I/O ภายในของ Z-80 PIO นี้ ประกอบด้วยพอร์ท 2 ชุด คือ พอร์ท A และ พอร์ท B และการเลือกพอร์ทต่าง ๆ ภายใน PIO กระทำได้โดยการใช้แอดเดรสบัส A<sub>0</sub> และ A<sub>1</sub> ต่อเข้ากับขา B/ $\bar{A}$  และ C/ $\bar{D}$  ของ PIO ความล่าช้าโดย B/ $\bar{A}$  จะทำหน้าที่เลือกพอร์ท ถ้าเป็นลอจิกเป็น "0" เลือกพอร์ท A และลอจิกเป็น "1" เลือกพอร์ท B ส่วน C/ $\bar{D}$  ทำหน้าที่เลือกการใช้งานของพอร์ท ถ้าลอจิกเป็น "0" เป็นการเลือกหน่วยข้อมูล และถ้าเป็น "1" จะเป็นการเลือกหน่วยควบคุม

การใช้งานของ Z-80 PIO ยังสามารถแบ่งออกเป็นหลายลักษณะ (Mode) แล้วแต่ความเหมาะสมในการใช้งานดังนี้

- Mode 0 เอาต์พุตแบบตรวจสอบสัญญาณ (Handshaking)
- Mode 1 อินพุตแบบตรวจสอบสัญญาณ



รูปที่ 3.7 แสดงขาสัญญาณต่าง ๆ ของ Z-80 PIO

Mode 2 อินพุต-เอาต์พุต สองทิศทาง (Bidirectional) แบบตรวจสอบสัญญาณ

Mode 4 ใช้พอร์ท A และพอร์ท B เป็นการควบคุม

สำหรับการนำมาใช้งานได้เลือกการทำงานใน Mode 3 โดยให้ PA0 เป็นอินพุต สำหรับรับสัญญาณลอจิก TTL จากหัวอ่านบาร์โค้ด และให้ PB0 เป็นเอาต์พุตสำหรับไปใช้ในการ ขับ LED เพื่อบอกการทำงาน 2 ลักษณะโดยใช้บอกสถานะการเริ่มต้นการทำงานเมื่อเริ่มเปิดเครื่อง โดยจะติดสว่างอยู่ประมาณ 1.5 วินาทีและบอกความถูกต้องในการอ่านข้อมูล โดยจะติดสว่างอยู่ประมาณ 1.5 วินาที เช่นกัน

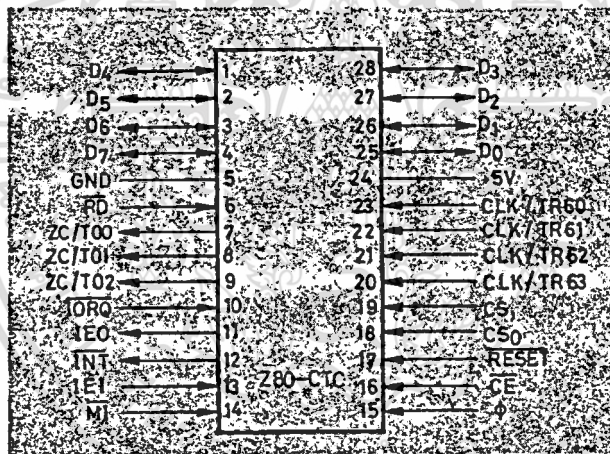
Z-80 CTC (Counter-Time Chip)

เป็นอุปกรณ์ใช้สำหรับเป็นวงจรมนับที่สามารถควบคุมได้โดยโปรแกรมโดยภายในมีวงจรมับอยู่ 4 ชุด แยกจากกัน ทั้ง 4 ชุด นี้สามารถโปรแกรมให้ทำงานได้ทั้งเป็นวงจรมับ (Counter) หรือวงจรมตั้งเวลา (Timer) ได้ ในแต่ละชุดเรียกว่า แชนแนล (Channel) โดยซีพียูใช้ CS0 และ CS1 เป็นตัวเลือกแชนแนลต่าง ๆ สำหรับการใช้งานของ CTC มีดังนี้

เป็นวงจรมตั้งเวลา (Timer Mode) โดยการทำสัญญาณ Clock ของระบบผ่าน Prescaler Register ซึ่งสามารถโปรแกรมให้หารด้วยค่า 16 หรือ 256 ได้ แล้วจึงป้อน

สัญญาณให้กับ Timer Constant Register อีกค่อหนึ่ง ใน Time Constant Register นี้สามารถโปรแกรมค่าคงที่เวลา (Time Constant) ขนาด 8 บิตได้ จากนั้น Down Counter จะเอาค่าTime Constantมาทำการนับ และเมื่อนับลงเป็น 0 ก็จะทำให้สัญญาณแอกทีฟ (Active) ออกมาที่ขา TO (Time Out)

เป็นวงจรมับ (Counter Mode) โดยการใช้สัญญาณอินพุตจากภายนอกมา Decrement ค่าคงที่ใน Counter Register (ซึ่งก็คือ Time Constant Register)โดยผ่านเข้ามาทางขา CLK/TRG สำหรับใน Mode นี้จะไม่มีการใช้ Prescaler Register และเช่นเดียวกับใน Timer Mode เมื่อทำการนับลงถึง 0 จะให้สัญญาณแอกทีฟออกมาที่ขา ZC (Zero Count) ซึ่งก็คือขาเดียวกับ TO



รูปที่ 3.8 แสดงขาสัญญาณต่าง ๆ ของ CTC

สำหรับการนำมาใช้งานในระบบนี้ ได้โปรแกรมให้แชนแนล 0 เป็นวงจรถัดเวลาโดยนำสัญญาณ Clock ของระบบ (2 MHz) ผ่าน Prescaler Register ซึ่งตั้งไว้ให้หาร 16 และให้แชนแนล 1 เป็นวงจรมับโดยการรับอินพุตจาก TO ของ แชนแนล 0 ดังนั้นเราจะได้ 16 บิต Down Counter โดยมีสัญญาณ Clock ที่ใช้ในการนับเท่ากับ 2 MHz หาร 16 = 125 KHz หรือเท่ากับใช้เวลา 0.008 ms ต่อการนับ 1 ครั้ง โดยจะนำไปใช้ในการนับเพื่อหาขนาดของมาร์แต่ละอัน

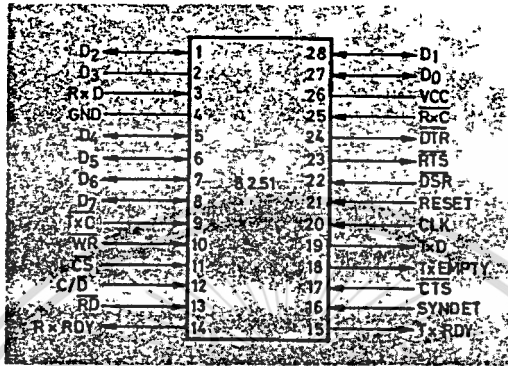
8251 - USART (Universal Synchronous/Asynchronous Receiver/Transmitter)

เป็น LSI ชิพทำหน้าที่เป็นตัวรับ และส่งข้อมูลแบบขนานกับไมโครโปรเซสเซอร์และรับ-ส่ง ข้อมูลแบบอนุกรมกับอุปกรณ์ภายนอกสามารถทำงานได้ทั้ง 2 ระบบ คือแบบ ซิงโครนัสและอซิโครนัส 8251-USART มีขาสัญญาณต่าง ๆ ดังแสดงในรูปที่ 3.9 การใช้งาน 8251 ในระบบนี้ให้รับข้อมูลแบบขนานจากซีพียู และส่งข้อมูลแบบอนุกรมในระบบของซิงโครนัส ไปยังพอร์ทอินพุทของไมโครคอมพิวเตอร์โดยลักษณะข้อมูลแบบอนุกรมของ 8251 ในระบบของซิงโครนัสแต่ละข้อมูลประกอบด้วย

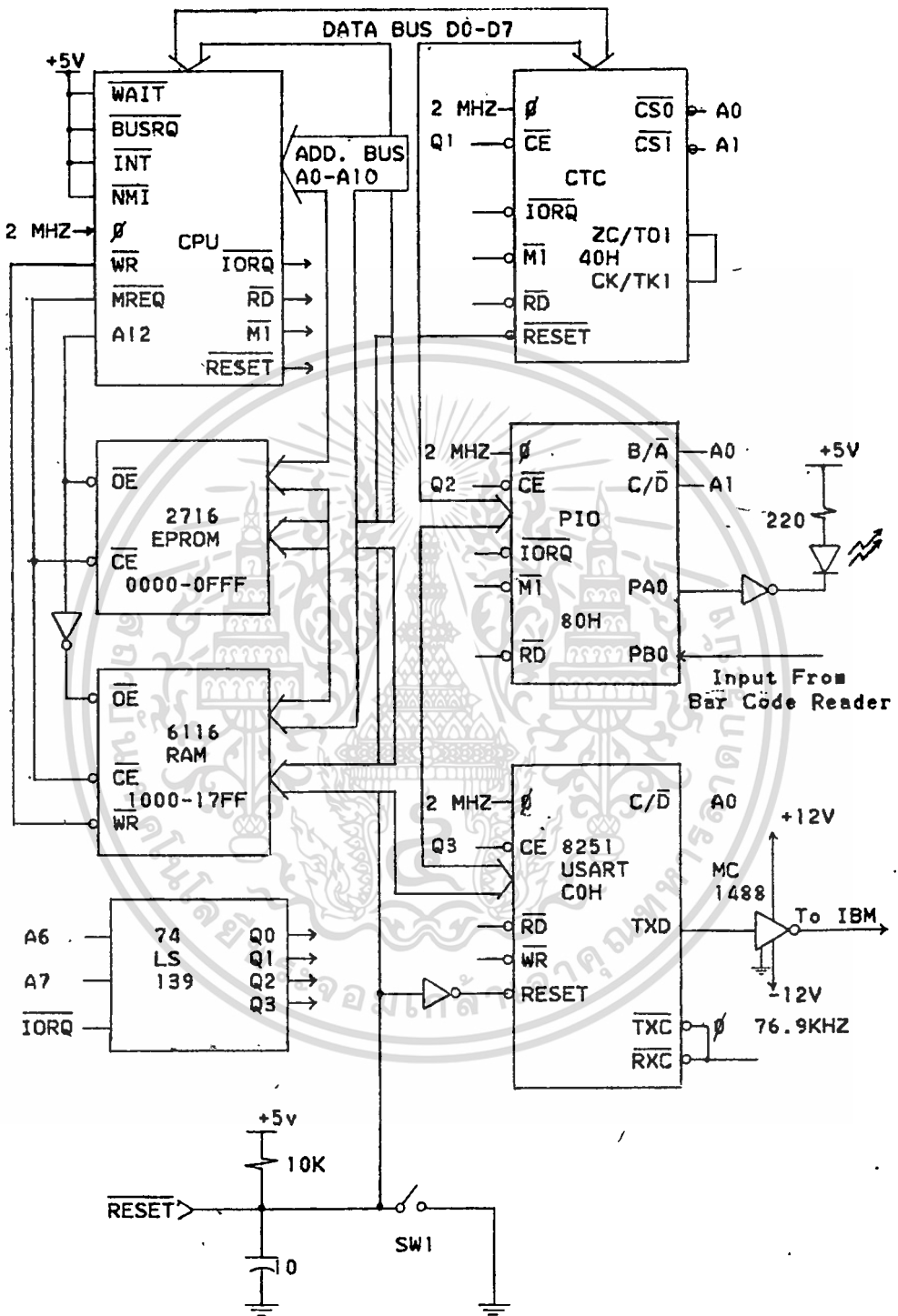
- รหัสหลายบิตเรียงต่อกันความยาวของข้อมูล เลือกได้ตั้งแต่ 5 ถึง 8 บิต
- Start Bit 1 บิต คือเป็นจุดเริ่มต้นของข้อมูล
- Parity Bit 1 บิต เป็นรหัสเพิ่มเติมเพื่อการตรวจสอบความถูกต้องของตัวรหัสในข้อแรก
- Stop Bit จำนวนตั้งแต่ 1 บิต ถึง 2 บิต เลือกได้โดยโปรแกรม

สำหรับสัญญาณนาฬิกาที่ใช้เป็นฐานเวลาของข้อมูลที่ส่งข้อมูลออกไป (Transmitter Clock) ในระบบของซิงโครนัสค่าความถี่ของ  $\overline{TXC}$  จะสูงเป็น 1,16 และ 64 เท่าของอัตราส่งจริง (Baud Rate) ซึ่งสามารถเลือกได้โดยโปรแกรม โดยการนำมาใช้ในระบบนี้ได้เลือกแบบ 16 เท่า และเลือก Baud Rate เป็น 4800 Baud ดังนั้นความถี่ของ Transmitter Clock ที่ป้อนเข้าทาง  $\overline{TXC}$  จะต้องเป็น  $4800 \times 16$  เท่ากับ 76.9 kHz ซึ่งได้จากวงจรสัญญาณนาฬิกาชุดที่ 2 ของระบบ ส่วนความยาวของข้อมูลได้เลือกแบบ 8 บิต และไม่มี Parity Bit

สำหรับวงจรของระบบไมโครโปรเซสเซอร์ที่ออกแบบขึ้นมีรายละเอียดแสดงในรูปที่



รูปที่ 3.9 แสดงขาสัญญาณต่างของ 8251 USART



รูปที่ 3.10 แสดงวงจรระบบไมโครโปรเซสเซอร์ที่ออกแบบขึ้น

#### บทที่ 4

### การสร้างโปรแกรมสำหรับเครื่องอ่านบาร์โคด

โปรแกรมที่สร้างขึ้นสำหรับใช้ในเครื่องอ่านบาร์โคดซึ่งควบคุมการทำงานด้วยระบบไมโครโปรเซสเซอร์เพื่อกำหนดขั้นตอนการทำงานต่าง ๆ ให้กับซีพียูในการอ่านสัญญาณบาร์โคดและตรวจสอบ จากนั้นจึงส่งข้อมูลไปยังไมโครคอมพิวเตอร์เพื่อนำไปประมวลผลต่อไป ในการออกแบบโปรแกรมจำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างของบาร์โคดพอสมควร โดยในบทนี้จะกล่าวถึงแนวการออกแบบและ ขั้นตอนการทำงานของโปรแกรม

#### 4.1 การกำหนดมาตรฐานในการอ่านบาร์โคด

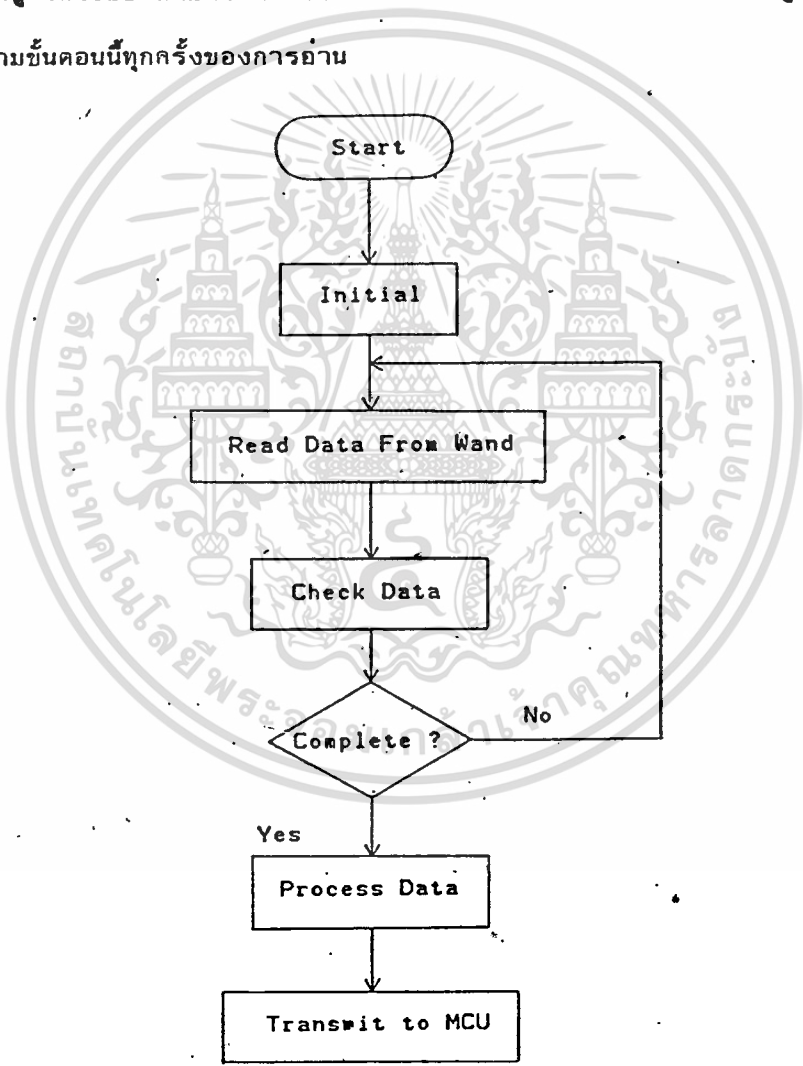
บาร์โคดที่ออกแบบขึ้นเพื่อใช้เป็นรหัส เป็นแบบ 6 ตัวเลขโดยแต่ละตัวเลขจะถูกแทนด้วย บาร์ดำ 2 บาร์ และบาร์ขาว 2 บาร์ ดังได้กล่าวมาแล้วในบทก่อน รวมทั้ง Guard Bar ทางด้านซ้ายและทางขวา ซึ่งเมื่อรวมจำนวนบาร์ทั้งหมดใน 1 รหัสของบาร์โคดจะประกอบด้วยบาร์ดำ 17 บาร์ และบาร์ขาว 16 บาร์ โดยจำนวนตัวเลขของบาร์ดำนี้จะถูกนำไปใช้ในการนับบาร์สำหรับโปรแกรมในการอ่าน

ในการใช้งานเครื่องอ่านบาร์โคด เจ้าหน้าที่ต้องจับหัวอ่านลากผ่านไปบนบาร์โคด ซึ่งอัตราการลากหัวอ่านผ่านไปบนบาร์โคด (Scanning Rate) นี้ มีสมมุติฐานว่าอัตราลากที่ช้าที่สุดที่ควรจะเป็นมีค่าไม่เกิน 0.5 นิ้ว ต่อวินาที และอัตราการที่เร็วที่สุดที่ควรจะเป็นมีค่าไม่เกิน 50 นิ้ว ต่อวินาที สำหรับกรณีที่อาจเกิดขึ้นในการใช้งานความเป็นไปได้ในการลากช้ากว่าปกติมีโอกาสเกิดขึ้นได้ โดยสาเหตุอาจมาจากการสะดุดในระหว่างการลากดังนั้นจึงจำเป็นต้องกำหนดค่าเวลาอ้างอิงขึ้นเพื่อใช้สำหรับการตรวจสอบการอ่านบาร์ โดยใช้ขนาดโมดูลของ UPC เป็นหลักซึ่งมีขนาดเท่ากับ 0.013 นิ้ว และอัตราการลากหัวอ่านเท่ากับ 0.5 นิ้ว ต่อวินาที ดังนั้นจะได้ค่าเวลานานที่สุดคือ 1 โมดูล เท่ากับ 0.013หารด้วย 0.5 = 0.026 วินาที ซึ่งจะกำหนดไว้ให้เป็นค่า  $t_m$  โดยที่เมื่อเทียบกับการนับของ Z-80 CTC ซึ่งมีคาบเวลาในการนับ 1 ครั้ง เท่ากับ 0.008 ms ก็จะได้ค่า  $t_m = \frac{0.026}{8 \cdot 10^{-6}} = 3250$  จากตารางสัญลักษณ์บาร์โคดของ UPC ในรูปที่ 1.4 จะพบว่า

ขนาดของบาร์ที่ใหญ่ที่สุดในแต่ละ Character จะมีขนาดเท่ากับ 4 ไมโครดั่งนั้นค่าคาบเวลาที่นานที่สุดที่ยอมได้ใน 1 บาร์ มีค่าเท่ากับ  $4 \times tm$  หรือ  $4 \times 0.026$  วินาที หรือ  $4 \times 3250$  ครั้ง ในการนับซึ่งจะนำไปใช้เป็นค่าอ้างอิงในการอ่านบาร์แต่ละบาร์ในโปรแกรม

#### 4.2 ขั้นตอนการทำงานของโปรแกรมควบคุม

การทำงานของโปรแกรมควบคุมสำหรับเครื่องอ่านบาร์โค้ด ซึ่งเป็นโปรแกรมที่แสดงการทำงานของ ซีพียู ในระบบ สามารถแสดงเป็น Flow Chart การทำงาน ดังในรูปที่ 4.1 ซึ่ง ซีพียู จะทำงานตามขั้นตอนนี้ทุกครั้งของการอ่าน



รูปที่ 4.1 แสดงการทำงานของโปรแกรมควบคุม

#### 4.3 รายละเอียดเกี่ยวกับการทำงานของโปรแกรมต่าง ๆ ในระบบ

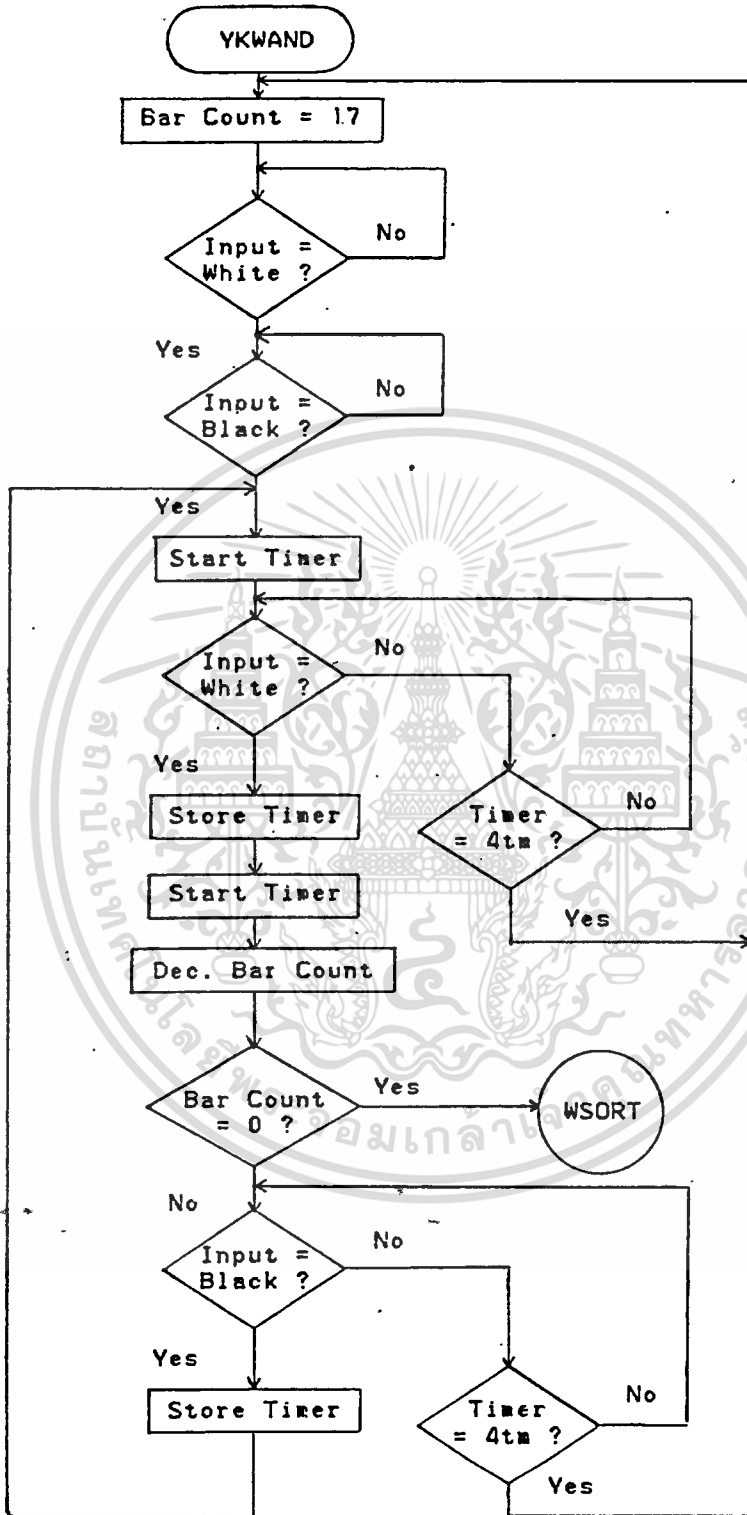
จากโปรแกรมหลักหรือโปรแกรมควบคุมสามารถแบ่งออกเป็นส่วนของโปรแกรมน้อยได้ตามลำดับหน้าที่การทำงาน ซึ่งมีรายละเอียดและขั้นตอนในการทำงานดังต่อไปนี้

##### 4.3.1 การทำงานของโปรแกรมที่ใช้ในการอ่านและตรวจสอบ (YKWAND)

หน้าที่ของโปรแกรมในส่วนนี้จะเริ่มด้วยการ Initial อุปกรณ์ต่าง ๆ ในระบบไมโครโปรเซสเซอร์ ซึ่งได้แก่ Z-80 PIO, Z-80 CTC, 8251 USART และการให้สัญญาณไฟของ LED ดิสสว่าง เพื่อแสดงความพร้อมของระบบ จากนั้นจึงอ่านข้อมูลเข้ามาจากหัวอ่าน (Wand) และทำการตรวจสอบข้อมูล โดยการตรวจค่าเวลาที่ใช้ในการอ่านของแต่ละบาร์กับค่าเวลาอ้างอิง ( $4 \times t_m$ ) และตรวจจำนวนบาร์ที่อ่านเข้ามาถ้ามีความผิดพลาดเกิดขึ้น โปรแกรมก็จะกลับไปเริ่มต้นทำการอ่านใหม่ทันที เมื่ออ่านข้อมูลเข้ามาเรียบร้อยแล้วก็จะถูกนำไปเก็บไว้ในหน่วยความจำ เพื่อนำไปประมวลผลต่อไป สำหรับ Flow Chart แสดงขั้นตอนการทำงานของโปรแกรมนี้ได้แสดงในรูปที่ 4.2

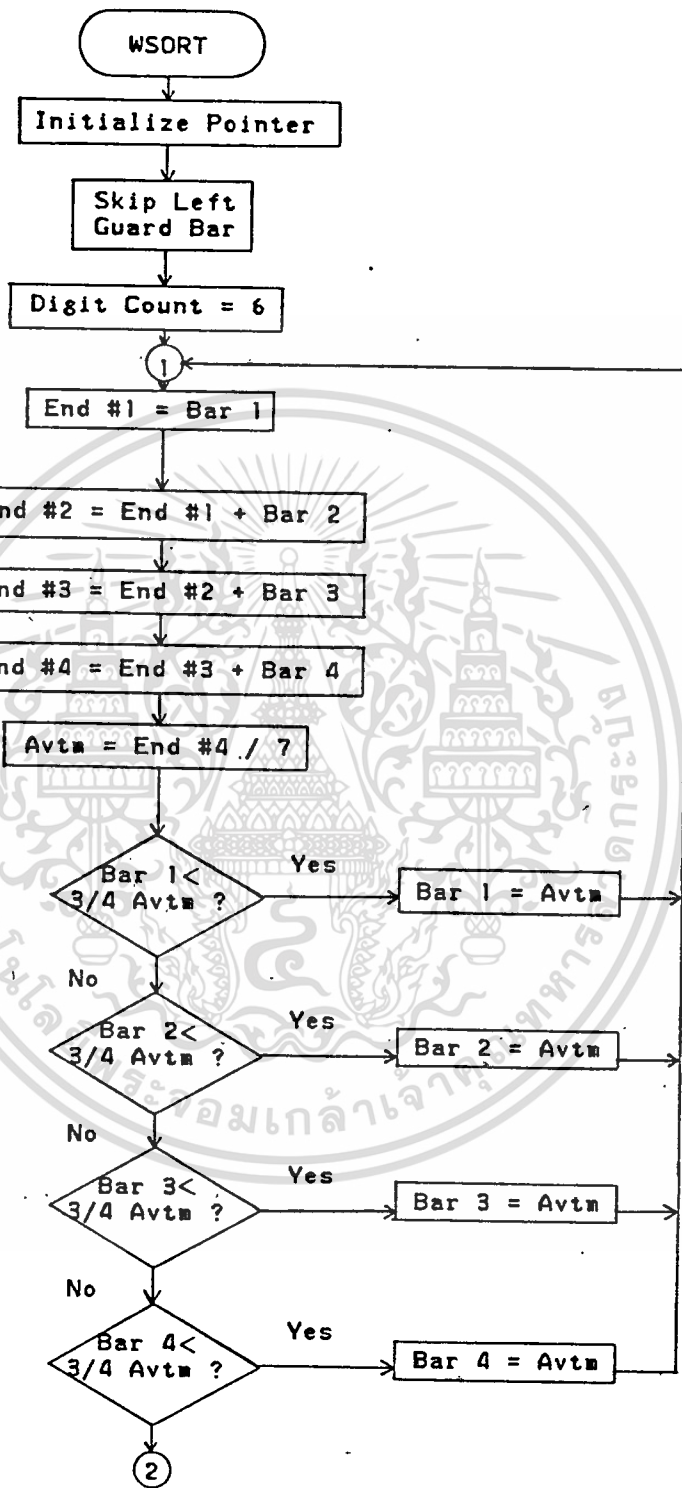
##### 4.3.2 การทำงานของโปรแกรมสำหรับ เปลี่ยนข้อมูลของบาร์ให้เป็นเลขไบนารี (WSORT)

การทำงานของโปรแกรมในส่วนนี้ เริ่มด้วยการนำข้อมูลของบาร์แต่ละบาร์ในหน่วยความจำที่ได้จากโปรแกรม YKWAND มาทำการปรับขนาดของบาร์โดยตรวจดูว่ามีบาร์ใดที่มีขนาดเล็กกว่า 3 ใน 4 ของขนาดโมดูลเฉลี่ย (Average Time Module หรือ Avtm) หรือไม่ ซึ่งค่าโมดูลเฉลี่ยนี้ได้จากการนำเอาบาร์ทั้งหมด ใน 1 Character (2 บาร์ดำ, 2 บาร์ขาว) มารวมคาบเวลาทั้งหมดหารด้วย 7 (1 Character Bar มี 7 โมดูล) ซึ่งถ้าพบว่า มีบาร์ใดที่มีขนาดเล็กเกินที่กำหนดไว้ ก็จะทำให้การแทนค่าบาร์นั้นด้วยค่าโมดูลเฉลี่ยทันที จุดประสงค์เพื่อเวลาอ่านข้อมูลของแต่ละบาร์จะสามารถทำการอ่าน ณ ตำแหน่งที่ใกล้เคียงจุดกึ่งกลางของข้อมูล (กึ่งกลางโมดูล) ซึ่งเป็นการอ่านข้อมูลที่จะให้ความถูกต้องมากที่สุด จากนั้นจึงทำการเปลี่ยนข้อมูลที่อ่านได้ให้อยู่ในรูปของเลขไบนารี (Binary Number) เพื่อนำไปตรวจสอบต่อไป สำหรับ Flow Chart แสดงการทำงานของโปรแกรมแสดงในรูปที่ 4.3

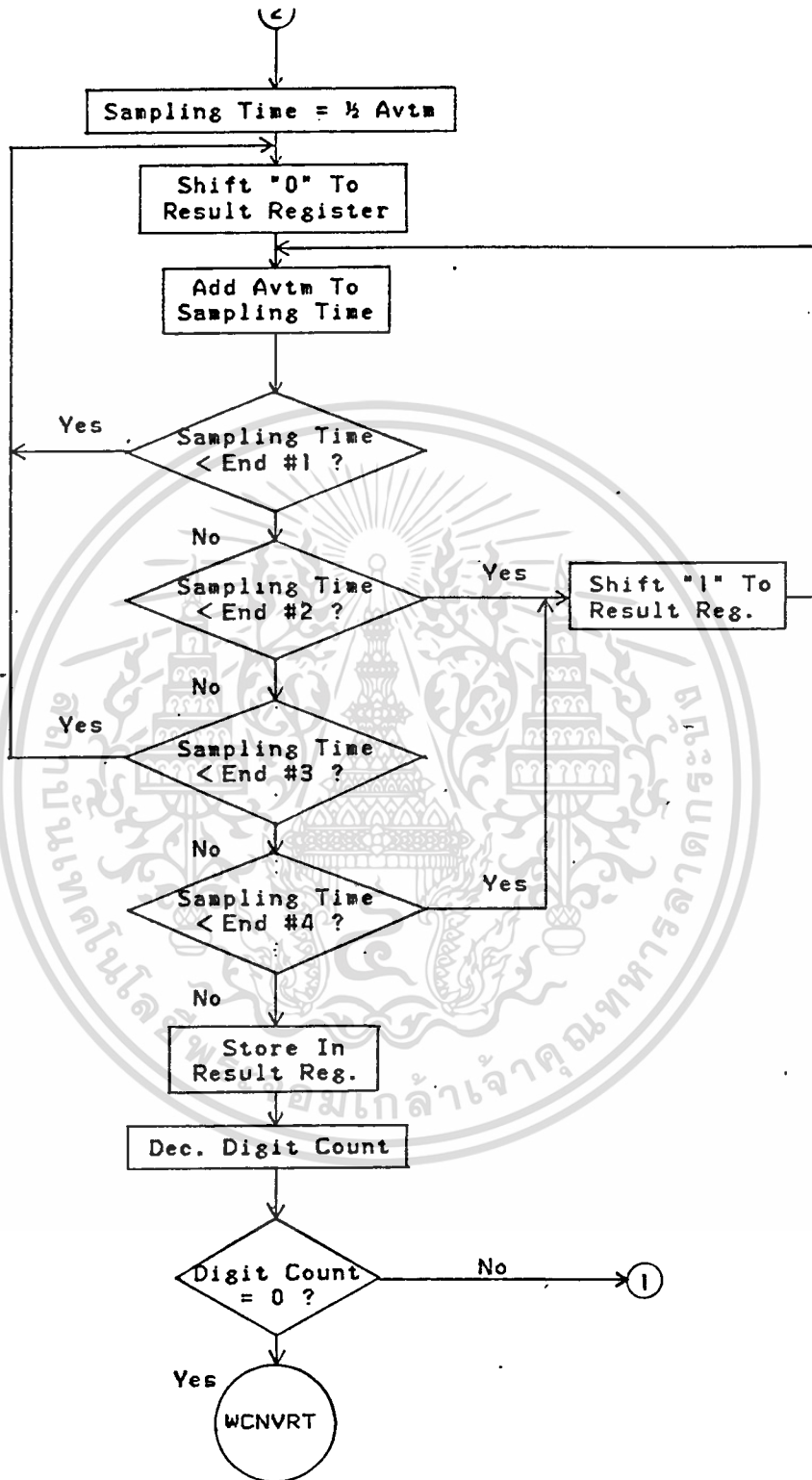


รูปที่ 4.2 แสดง Flow Chart ของ "YKWAND"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดง Flow Chart ของ "WSORT"



รูปที่ 4.3 (ต่อ)

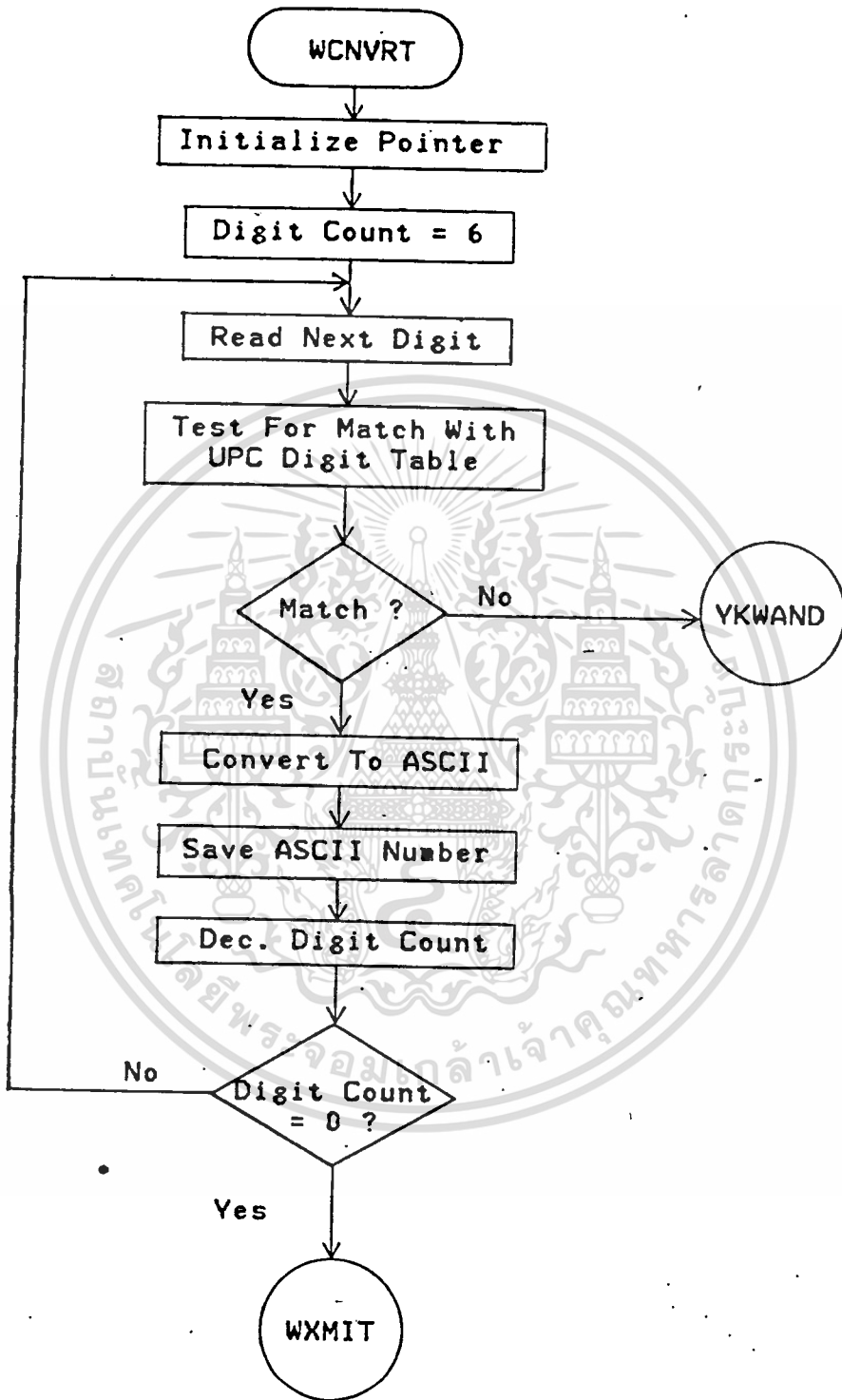
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. 3.3 การทำงานของโปรแกรมในการตรวจสอบข้อมูลไบนารีและ เปลี่ยน เป็นรหัส ASCII

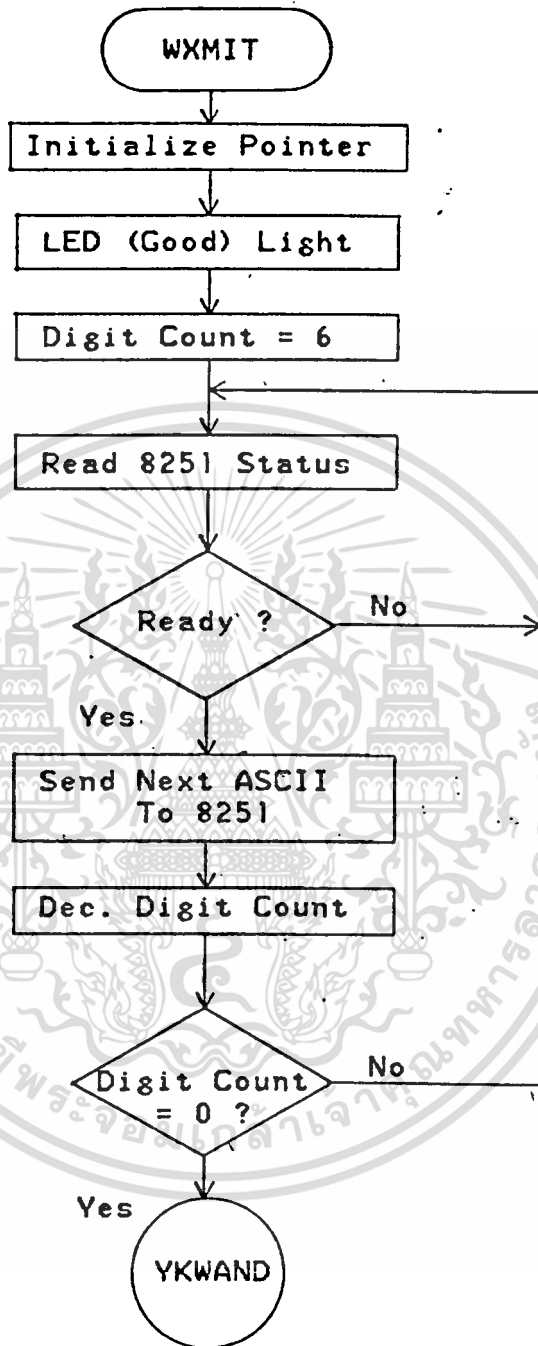
การทำงานของโปรแกรมส่วนนี้จะนำข้อมูลของแต่ละ Character ในรูปของเลขไบนารี มาทำการเปรียบเทียบกับ UPC Digit Table ซึ่งเก็บสัญลักษณ์ของ UPC Character ในรูปของไบนารีไว้ และถ้าหาไม่พบ โปรแกรมก็จะกลับไปเริ่มต้นทำการอ่านข้อมูลใหม่ในโปรแกรม YKWAND และเมื่อทำการตรวจสอบเรียบร้อยแล้วก็จะเปลี่ยนข้อมูลที่ได้อ่านอยู่ในรูปของ รหัส ASCII จากนั้นจึงส่งไปยังโปรแกรมส่วนต่อไปสำหรับ Flow Chart การทำงานได้แสดงในรูปที่ 4.4

#### 4. 3.4 การทำงานของโปรแกรมสำหรับส่งข้อมูล (WXMIT)

โปรแกรมส่วนนี้จะทำหน้าที่ส่งข้อมูลของบาร์โค้ดในรูปของรหัส ASCII จากระบบไมโครโปรเซสเซอร์ ไม่งทอร์ทอินพุตแบบอนุกรม RS-232C ของไมโครคอมพิวเตอร์ โดยผ่านทาง 8251 USART และการให้สัญญาณไปของ LED ดิจสว่าง เพื่อบอกความถูกต้องในการอ่านข้อมูล Flow Chart แสดงการทำงานได้แสดงไว้ในรูปที่ 4.5



รูปที่ 4.4 แสดง Flow Chart ของ "WCNVRT"



รูปที่ 4.5 แสดง Flow Chart ของ "WXMIT"

## บทที่ 5

### การใช้ไมโครคอมพิวเตอร์ในการพัฒนาระบบห้องสมุด

ในการทำงานของห้องสมุดที่ห้องสมุด โดยทั่วไปมัก เป็นงานที่ต้องกระทำซ้ำ ๆ กัน เช่น งานบริการในด้านการยืม และการรับคืนหนังสือ ซึ่งเป็นงานบริการที่เจ้าหน้าที่ติดต่อกับผู้ใช้บริการ มากงานหนึ่ง ในลักษณะของงานดังกล่าวต้องจัดเตรียมขั้นตอนต่าง ๆ ซึ่งได้แก่การทำบัตรรายการ ประจำหนังสือ และบัตรบันทึกรายการประจำตัวผู้ใช้บริการ เป็นต้น การบันทึกเปลี่ยนแปลงข้อมูล ในบัตรเหล่านี้จะเกิดขึ้นตลอดเวลาที่มีการใช้บริการ ดังนั้นการนำเอาระบบไมโครคอมพิวเตอร์มา ใช้ในการทำโปรแกรมสำหรับในห้องสมุด โดยร่วมกับการใช้ รหัสบาร์โคดจึงเป็นการเพิ่มประสิทธิภาพและความสะดวกในการทำงานของเจ้าหน้าที่

#### 5.1 ระบบงานบริการทั่วไปในห้องสมุด

โดยปกติงานในห้องสมุดที่เจ้าหน้าที่ให้บริการแก่ผู้คนที่ทั่วไป นอกเหนือจากงานประจำที่ต้องทำอยู่แล้ว เช่น การจัดเก็บหนังสือ การจัดซื้อหนังสือ และการทำบันทึกประจำหนังสือ เป็นต้น พหุจำแนกประเภทของงานบริการได้ดังนี้

- การยืมหนังสือ
- การคืนหนังสือ
- การตรวจสอบหนังสือว่าอยู่ในห้องสมุดหรือไม่
- การตรวจสอบหนังสือครบกำหนดส่งคืน
- การบริการจองหนังสือ

ซึ่งงานในลักษณะดังกล่าว เจ้าหน้าที่ต้องใช้บัตรทำรายการต่าง ๆ เช่น บัตรบันทึก รายการประจำตัวผู้ใช้บริการ ซึ่งจะใช้บันทึกรายละเอียดเกี่ยวกับเลขหมายของหนังสือที่ได้ยืมไป รวมทั้งวันที่ยืมและวันส่งคืนหนังสือ ส่วนบัตรประจำหนังสือจะใช้บันทึกชื่อผู้ยืม และวันส่งคืนหนังสือ ในการปฏิบัติงานของเจ้าหน้าที่ ต้องทำการจดบันทึกรายละเอียดเหล่านี้ทุกครั้งที่มีผู้ใช้บริการ นอกจากนี้เจ้าหน้าที่ต้องเก็บรักษาบัตรบันทึกรายการประจำหนังสือไว้ เมื่อมีการยืมหนังสือ ในการ

ปฏิบัติงานของเจ้าหน้าที่ ต้องทำการจัดบันทึกรายละเอียดเหล่านี้ทุกครั้งที่มิผู้ใช้บริการ นอกจากนี้ เจ้าหน้าที่ต้องเก็บรักษาบัตรบันทึกรายการประจำหนังสือไว้ เมื่อมีการยืมหนังสือ ในการคืนหนังสือ ก็เช่นกัน เจ้าหน้าที่ต้องตรวจดูว่าคืนหนังสือถูกต้องหรือไม่ และทำการบันทึกวันที่คืนลงในบัตรทั้งสองพร้อมทั้งตรวจดูว่าได้เลยกำหนดวันส่งคืนหรือไม่ เพื่อทำการเก็บค่าปรับ

นอกจากนี้ในการค้นหาหนังสือเล่มใดว่าอยู่ในห้องสมุดหรือไม่คือได้ถูกยืมไปหรือไม่เจ้าหน้าที่ก็ต้องทำการตรวจจากบัตรบันทึกรายการของหนังสือเล่มนั้นว่ามีในส่วนของหนังสือที่ถูกยืมไปหรือไม่ ถ้าอยู่ที่แสดงว่าได้ถูกยืมไปแล้ว ในการปฏิบัติงานดังที่ได้กล่าวมานี้ จะเห็นได้ว่ามีขั้นตอนที่ยุ่งยากพอสมควรซึ่งถ้ามีการใช้บริการเป็นจำนวนมาก อาจเกิดความล่าช้า ผิดพลาด และอาจเกิดการสูญหายของบัตรบันทึกรายการได้

## 5.2 ระบบไมโครคอมพิวเตอร์

เนื่องจากโปรแกรมห้องสมุดนี้ได้พัฒนาขึ้นโดยใช้ Package ของ DBASE III - Plus บน Disk Operating System (DOS) ของ IBM PC ซึ่งเป็นไมโครคอมพิวเตอร์ขนาด 16 บิต (Bit) ดังนั้นไมโครคอมพิวเตอร์ขนาด 16 บิต ทั่วไปที่สามารถ Compatible กับเครื่อง IBM PC, XT ก็สามารรถที่จะนำมาใช้แทนกันได้ สำหรับข้อมูลทั่วไปของเครื่องมีดังนี้

อุปกรณ์ฮาร์ดแวร์	ไมโครโปรเซสเซอร์ Intel 8088
	โปรเซสเซอร์ร่วม Intel 8087
สัญญาณนาฬิกา	4.77 MHz
ขนาดหน่วยความจำ ROM	40 K Byte
	RAM 256 K Byte ขยายได้ถึง 640 K Byte
คีย์บอร์ด	83 คีย์ มีฟังก์ชันคีย์ 10 คีย์ และมีคีย์ตัวเลขแยกต่างหาก
หน่วยความจำสำรอง ฟลอปปีดิสก์ (Floppy Disk)	แบบ 2 หน้า
	ความจุ 360 K Byte
ช่องขยาย	มีสล롯 (Slot) 8 สล롯 ใช้ 3 สลอต การ์ดสำหรับควม

คุณภาพและ ความคุมการสื่อสารข้อมูลแบบ อะซิงโครนัส  
(Asynchronous) RS-232 C

ซอฟต์แวร์ใน ROM มีไบออส (Bios) โปรแกรมตรวจสอบระบบและ เมลิก

สำหรับรายละเอียดเกี่ยวกับ โครงสร้างภายในมีดังนี้

5.2.1 โครงสร้าง เมนบอร์ดของเครื่อง IBM PC,XT

โครงสร้างของเมนบอร์ด เป็นแผ่นพีซีบอร์ดชนิดหลายชั้น (Multilayer PCB) การจัดตำแหน่งของไอซีต่าง ๆ บนบอร์ดและรูปเมนบอร์ดแสดงดังรูปที่ 5.1 บนบอร์ดประกอบด้วยส่วนวงจรที่มีไมโครโปรเซสเซอร์ 8088 เป็นหลักและมีซ็อกเก็ต (Socket) ของ 8087 ที่จะ เป็นโปรเซสเซอร์ร่วมในการทำงานทางคณิตศาสตร์ให้มีความเร็วยิ่งขึ้น นอกจากนี้บนบอร์ดจะมี ROM 40 กิโลไบต์มี RAM 256 กิโลไบต์ ขยายได้ถึง 640 กิโลไบต์ โดยจัดเป็นแถวละ 64 กิโลไบต์ มีซ็อกเก็ตสลอต (Socket Slot) แบบ 62 ขาจำนวน 8 ช่อง สำหรับเครื่องโยงกับระบบฮาร์ดแวร์อื่น สล็อตจะมีโครงสร้างสัญญาณเหมือนกันยกเว้นสลอต J8 ซึ่งมีโครงสร้างการจัดสัญญาณแตกต่างกับสลอตอื่น

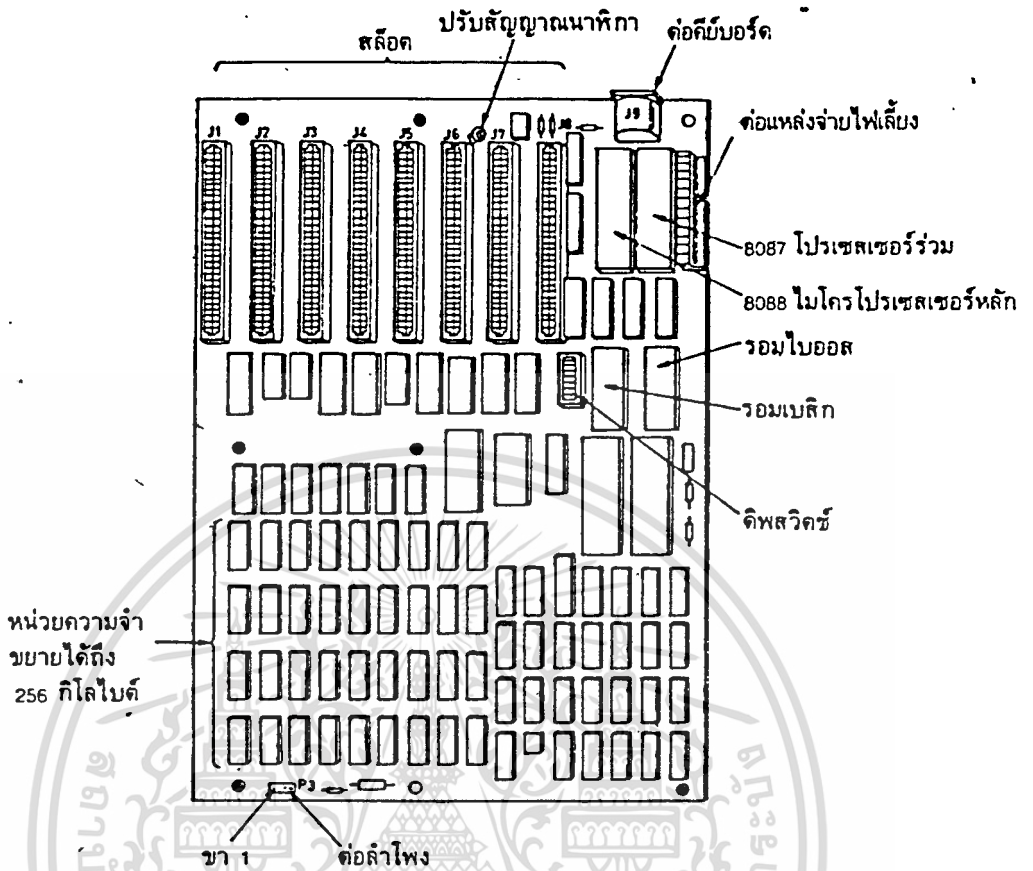
5.2.2 ไมโครโปรเซสเซอร์ 8088

ไมโครโปรเซสเซอร์ 8088 เป็น ซีพียู ในตระกูลของ 8086 ใช้บัสข้อมูลขนาด 8 บิต แอคเตสบัส 20 เส้น จึงสามารถใช้กับหน่วยความจำได้ถึง 1 เมกะไบต์ ( $2^{20} = 1048576$ ) การอ่านข้อมูลหรือเขียนข้อมูลจากหน่วยความจำจะอ่านทีละ 8 บิต แต่การประมวลภายในจะกระทำทีละ 16 บิต สัญญาณนาฬิกาที่ใช้มีความถี่ 4.77 เมกะเฮิรตซ์ วงจรประกอบไมโครโปรเซสเซอร์ 8088 แสดงดังรูปที่ 5.2

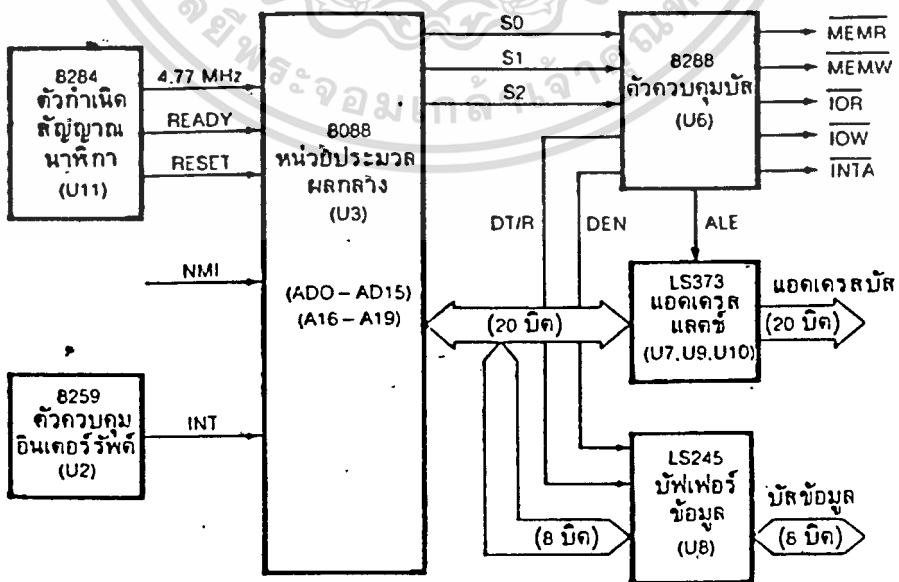
5.2.3 การจัดโครงสร้างบัสบนไมโครคอมพิวเตอร์

โครงสร้างบัสที่ออกจาก ซีพียู จะแยกออกเป็น 3 กลุ่มคือ บัสข้อมูล แอคเตสบัส และบัสควบคุม โดยบัสข้อมูลและบัสควบคุมมีขนาด 8 บิต ส่วน แอคเตสบัสมีทั้งเส้น 20 บิต ระบบบัสทั้งหมดของเครื่องไอเอ็มพีซีทุกส่วน จะมีการผ่านบัฟเฟอร์อีกครั้ง โครงสร้างของบัสเมื่อเขียนเป็นโคแอมแกรมได้ดังรูปที่ 5.3

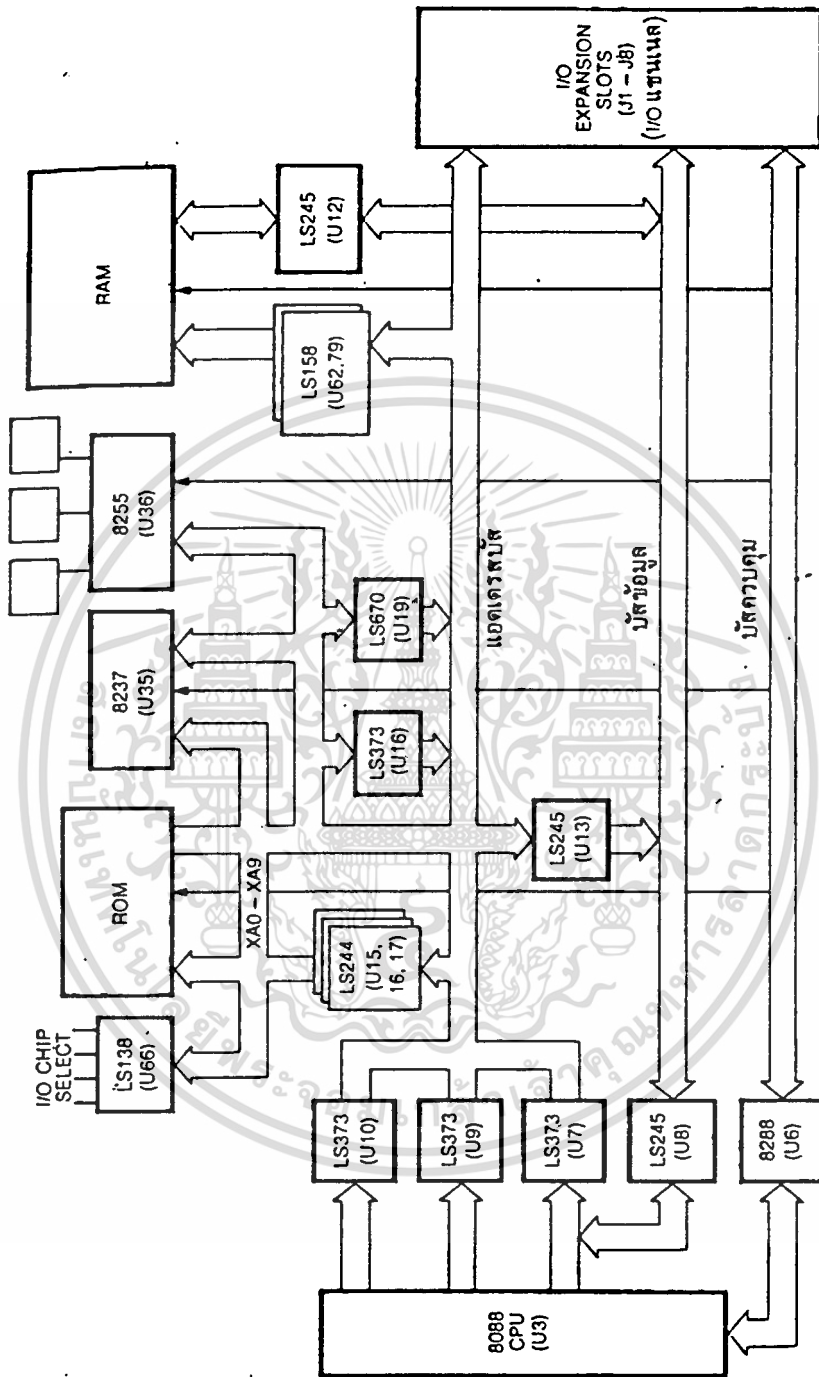
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 แสดงการจัดวางอุปกรณ์บนบอร์ดของ IBM PC, XT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งหา **รูปที่ 5.2** โครงสร้างไมโครโปรเซสเซอร์ 8088 เอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แสดงโครงสร้างขั้วสมบนไมโครคอมพิวเตอร์

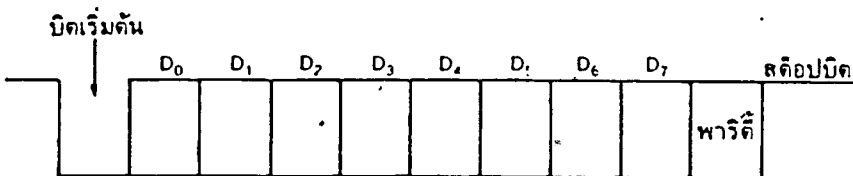
### 5.2.4 อะแดปเตอร์สื่อสารข้อมูล

บนไมโครคอมพิวเตอร์ มีระบบการติดต่อสื่อสารข้อมูลแบบอนุกรม โดยการกำหนด เป็นพอร์ท COM1 และ COM2 พอร์ททั้งสองนี้จะได้รับการโปรแกรมตามความต้องการของผู้ใช้งาน โดยส่วนอินเทอร์เฟซกับภายนอกจะเป็นไปตามมาตรฐาน RS-232 C

การติดต่อสื่อสารของข้อมูลแบบอนุกรมแบบที่นิยมใช้กันมากที่สุดคือแบบอะซิงโครนัสโดย มีส่วนของบิตเริ่มต้น (Start Bit) บิตสิ้นสุด (Stop Bit) บิตทั้งหมดจะประกอบเข้ามาเป็นเฟรม ในขณะที่ส่งรับข้อมูล วงจรฮาร์ดแวร์ที่ใช้ในการสื่อสารแบบอนุกรมนี้เป็นไอซีประเภท Asynchronous Receiver and Transmitter เบอร์ 8250 ซีพียู 8250 นี้สามารถโปรแกรมให้มีการทำงานตาม รูปแบบของการสื่อสารได้ ซึ่งเลือกอัตราการส่งได้ (Baud Rate) ตั้งแต่ 50 ถึง 9600 Baud มีสต่อบิตขนาด 1 ครั้ง หรือ 2 บิต บิตข้อมูลจะเป็นแบบ 5, 6, 7 หรือ 8 บิตก็ได้ ขีดความสามารถ ของบิตอะซิงโครนัสนี้ได้แก่

- การรับข้อมูล ไม่ขึ้นกับสัญญาณนาฬิกาที่ป้อนเข้ามาทางอินพุตและไม่ต้องซิงโครไนซ์ (Synchronize)
- มีส่วนควบคุม โมเด็ม (Modem)
- สามารถตรวจสอบ การผิดพลาดในการรับส่งข้อมูลได้
- มีขั้ว เฟอร์การรับส่งข้อมูล

การสื่อสารข้อมูลแบบ 8 บิต โดยมีบิต  $D_0$  เป็น MSB บิต คือส่งออกไปก่อน รูปแบบ การส่งเป็นดังรูปที่ 5.4



รูปที่ 5.4 แสดงสถานะของข้อมูลที่ส่งออก

## บทที่ 6

### การออกแบบและพัฒนาโปรแกรมห้องสมุด

โปรแกรมห้องสมุดที่ออกแบบและพัฒนาขึ้นนี้ ได้ออกแบบไว้ให้สามารถอ่านข้อมูลได้ทั้งจาก เครื่องอ่านบาร์โค้ดโดยผ่านทางพอร์ทอินพุทแบบอนุกรม RS-232 C หรือ อ่านจากคีย์บอร์ด (Keyboard) ของเครื่องไมโครคอมพิวเตอร์ ทั้งนี้เพื่อสำหรับในกรณีที่เครื่องอ่านบาร์โค้ดเกิดขัดข้อง เจ้าหน้าที่ก็ยังสามารถทำงานต่อไปได้

โปรแกรมห้องสมุดนี้ได้พัฒนาและสร้างขึ้นด้วย DBASE III - Plus ซึ่งเป็น Database Management Software Package แบบหนึ่ง ดังนั้นขั้นตอนในการออกแบบจึงต้องเริ่มด้วยการสร้างฐานข้อมูล (Database) ที่ต้องการใช้ขึ้นก่อนจากนั้นจึงออกแบบโปรแกรมเพื่อประมวลผล (Process) ข้อมูลในฐานข้อมูลที่ได้สร้างขึ้นตามการใช้งานที่ต้องการ สำหรับโปรแกรมที่ใช้ในการรับข้อมูลได้ถูกพัฒนาขึ้นด้วยภาษา แอสเซมบลี จากนั้นจึงเปลี่ยนให้อยู่ในรูปของ Binary File ซึ่งสามารถที่จะถูกเรียกใช้ได้โดย โปรแกรม DBASE III - Plus โดยฐานข้อมูลและโปรแกรมที่ใช้งานทั้งหมดจะถูก Load ใส่ Floppy Disk ใน Drive B.

#### 6.1 การออกแบบฐานข้อมูลสำหรับโปรแกรมห้องสมุด

ข้อมูลที่ใช้สำหรับการทำงานในระบบห้องสมุด สามารถแบ่งได้เป็น 2 ประเภท คือ ข้อมูลเกี่ยวกับหนังสือ และข้อมูลเกี่ยวกับผู้ใช้บริการดังนั้นในการออกแบบแฟ้มข้อมูลของฐานข้อมูล (Database File) จึงมีด้วยกัน 2 แฟ้ม โดยมีรายละเอียดดังต่อไปนี้

##### 6.1.1 แฟ้มข้อมูลหนังสือ (Book Database File) เป็นแฟ้มที่ใช้เก็บข้อมูล

เกี่ยวกับรายละเอียดของหนังสือแต่ละเล่ม ซึ่งประกอบด้วย

- เลขทะเบียนหนังสือ (Accession No.)
- เลขเรียกหนังสือ (Call No.)
- ชื่อผู้แต่ง (Author)
- ชื่อเรื่อง (Title)

นอกจากนี้ยังออกแบบไว้ใช้เก็บสถานะภาพของหนังสือแต่ละเล่มด้วยว่า อยู่ในห้องสมุดหรือได้ถูกยืมไป เลขรหัสประจำตัวผู้ยืม และ วันครบกำหนดส่งคืนหนังสือ โดยข้อมูลทั้งหมดเหล่านี้จะถูกเก็บอยู่ในลักษณะของ Filed ใน Record ของหนังสือแต่ละเล่ม สำหรับรายละเอียดของแฟ้มข้อมูลหนังสือได้จากภาคผนวก ในการค้นหาข้อมูลในแฟ้มหนังสือนี้ ได้ออกแบบให้ Field ของ เลขทะเบียนหนังสือ เลขเรียกหนังสือ ชื่อผู้แต่ง ชื่อเรื่อง และ วันครบกำหนดส่งคืน เป็นคีย์(Key) สำหรับใช้ในการค้นหาข้อมูล ซึ่งการจะเลือกใช้คีย์ใดในการค้นหานั้นขึ้นอยู่กับฟังก์ชันการใช้งานแต่ละประเภท

6.1.2 แฟ้มข้อมูลผู้ใช้บริการ (User Database File) เป็นแฟ้มที่ใช้เก็บข้อมูลเกี่ยวกับรายละเอียดของผู้ใช้บริการ ซึ่งประกอบด้วย

- เลขรหัสประจำตัว (Identification No.)
- ชื่อและนามสกุล (Name)
- สถานะภาพ (Status) (นักศึกษา, อาจารย์หรือเจ้าหน้าที่)
- คณะหรือสังกัด (Department)
- เลขทะเบียนของหนังสือที่ได้ยืมไป (Accession No.)
- จำนวนหนังสือที่มีสิทธิยืมได้

สำหรับรายละเอียดของแฟ้มข้อมูลผู้ใช้บริการได้จากภาคผนวก และในการค้นหาข้อมูลในแฟ้มนี้ ได้ออกแบบให้ใช้ Field ของ เลขรหัสประจำตัว เป็นคีย์ สำหรับใช้ในการค้นหาข้อมูล

## 6.2 การสร้างฐานข้อมูล

เนื่องจากการออกแบบระบบห้องสมุดได้ใช้ภาษา DBASE III - Plus ดังนั้นการสร้างโครงสร้าง (Schema) ของฐานข้อมูล ทั้ง 2 ประเภทจึงทำได้โดยการใช้คำสั่งของภาษานี้ โดยกำหนดให้ชื่อ File ของทั้ง 2 ประเภทคือ BOOKS. DBF เป็นชื่อ File ของหนังสือและ USER. DBF เป็นชื่อ File ของผู้ใช้บริการ โดยคำสั่งที่ใช้ในการสร้าง File มีรูปแบบทั่วไป

ดังนี้

```
. CREATE <database file name>
```

ตัวอย่างเช่น การสร้างโครงสร้างของฐานข้อมูลชื่อ "BOOKS . DBF" เก็บไว้ใน Dirve B มีดังนี้

```
. CREATE B: BOOKS
```

เมื่อใช้คำสั่งดังกล่าวแล้ว จอภาพของไมโครคอมพิวเตอร์จะแสดงเพื่อให้ผู้ใช้ป้อน ชื่อ ชนิด และขนาด ของ Field ต่าง ๆ ที่ได้ออกแบบไว้ในหัวข้อ 6.1 เมื่อป้อนรายละเอียดของ Field ต่าง ๆ ครบแล้วซอฟต์แวร์ DBASE . III - Plus จะสร้างโครงสร้างให้ฐานข้อมูลตามชื่อและรายละเอียดที่ได้กำหนดให้ หลังจากนั้นจึงทำการป้อนข้อมูลเข้าไปโดยแบ่งเป็นข้อมูลของหนังสือ และข้อมูลของผู้ใช้บริการเมื่อป้อนข้อมูลทั้งหมดเรียบร้อยแล้วให้ใช้คีย์ Ctrl END เพื่อบอกให้ DBASE III - Plus ทราบถึงการสิ้นสุดการป้อนข้อมูล และทำการเก็บข้อมูลทั้งหมดลงไปในแผ่นจากแม่เหล็ก (Diskkett) โดยใช้คำสั่ง

```
. USE
```

สำหรับการที่ผู้ใช้จะทำการเพิ่มเติมข้อมูลเข้าฐานข้อมูลในภายหลังเช่นการป้อนข้อมูลเข้า Database File ของ "BOOKS . DBF" จะใช้คำสั่งดังนี้

```
. USE B: BOOKS
```

```
. APPEND
```

เมื่อป้อนข้อมูลเรียบร้อยแล้วให้กระทำเช่นเดียวกับในกรณีการสร้างฐานข้อมูล

### 6.3 การสร้าง Index File ของแต่ละฐานข้อมูล

การสร้าง Index File ให้กับฐานข้อมูลนั้น จุดประสงค์เพื่อให้การ Retrive ข้อมูลจากฐานข้อมูลรวดเร็วขึ้นแทนการ Retrive ทีละ Record แบบ Sequential ซึ่งจะเสียเวลามาก สำหรับระบบห้องสมุดนี้ได้ออกแบบให้ฐานข้อมูลของหนังสือ (BOOKS . DBR) มี Field ที่ใช้เป็นคีย์ ในการสร้าง Index File อยู่ 5 Field และของผู้ใช้บริการ (USER

DBF) มี Field ที่ใช้เป็นคีย์อยู่ 1 Field (ดูรายละเอียดใน Document No: DBF-LIB01 ในภาคผนวก) การสร้าง Index File กระทำได้โดยใช้คำสั่งของ DBASE III - Plus ซึ่งมีรูปแบบของคำสั่งทั่วไปดังนี้

. INDEX ON <key field name> TO <index file name>

โดย key field name คือชื่อของ field ในฐานข้อมูลที่จะใช้เป็นคีย์

และ index file name คือชื่อของ Index File ที่สร้างขึ้น

#### 6.4 การออกแบบขั้นตอนการทำงานของโปรแกรม

เมื่อได้ฐานข้อมูลสำหรับระบบห้องสมุดเรียบร้อยแล้วต่อไปจะเป็นการสร้างโปรแกรมสำหรับใช้ในระบบห้องสมุด โดยโปรแกรมที่สร้างขึ้นก็คือ ชุดคำสั่งของ DBASE III - Plus เพื่อทำหน้าที่เรียกใช้ฐานข้อมูลที่เกี่ยวข้อง มาทำการแสดงข้อมูล หรือปรับปรุง แก้ไขข้อมูลตามแต่ละประเภทของการใช้งาน โดยโปรแกรมห้องสมุดนี้ได้ออกแบบให้มีลำดับขั้นตอนในการทำงานดังนี้

เริ่มต้นด้วยการเลือกประเภทของการทำงานโดยแบ่งออกเป็น 4 รายการคือ

1. การค้นหาหนังสือ (Search The Book)
2. การยืมหนังสือ (Borrow The Book)
3. การคืนหนังสือ (Return The Book)
4. การตรวจสอบหนังสือที่ครบกำหนดส่งคืน (Borrower Due Date)รายละเอียดของแต่ละรายการมีดังนี้

การค้นหาหนังสือ สามารถกระทำได้โดยเจ้าหน้าที่เลือกชนิดของการค้นหา

- คือ - เลขทะเบียนหนังสือ  
- เลขเรียกหนังสือ  
- ชื่อผู้แต่ง  
- ชื่อเรื่อง

โดยเจ้าหน้าที่จะใช้คีย์บอร์ดสำหรับการป้อนข้อมูล

การพิมพ์และการคืนหนังสือ ได้ออกแบบไว้สำหรับให้ใช้เลขทะเบียนหนังสือ และ เลขประจำตัวผู้ใช้บริการ เป็นข้อมูลในการทำงาน โดยเจ้าหน้าที่สามารถเลือกวิธีการป้อนข้อมูลได้ 2 วิธี คือ

- ป้อนข้อมูลโดยใช้เครื่องอ่านบาร์โคด
- ป้อนข้อมูลทางคีย์บอร์ด

โดย เริ่มต้นการทำงานด้วยการป้อนข้อมูลของผู้ใช้บริการก่อนจากนั้นจึงป้อนข้อมูลของหนังสือตามลำดับ

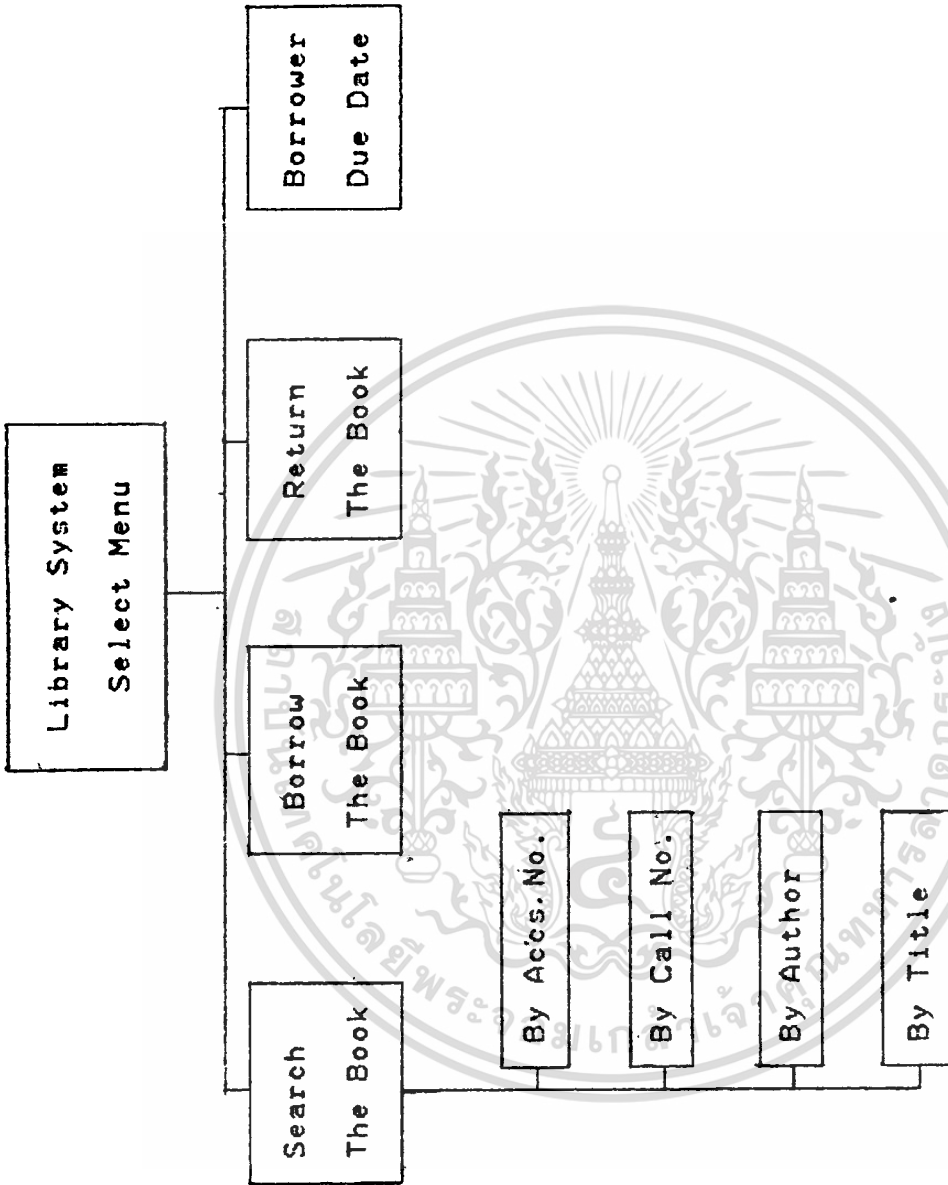
การตรวจสอบหนังสือที่ครบกำหนดส่งคืน สำหรับการทำงานในส่วนนี้ไม่มีการใช้ข้อมูลจากภายนอก เป็นเพียงแต่การ เรียกดูข้อมูลจากฐานข้อมูลภายใน เท่านั้น

สำหรับการแสดงการทำงานทางจอภาพของ เครื่องในการทำการต่าง ๆ ดูรายละเอียดได้ในภาคผนวก

ลำดับขั้นตอนการทำงานของ โปรแกรมในระบบห้องสมุดสามารถเขียน เป็นไดอะแกรม ได้ดังรูปที่ 6.1

#### 6.5 การสร้างโปรแกรมห้องสมุด

เมื่อได้ออกแบบฐานข้อมูลสำหรับใช้ในระบบห้องสมุด รวมทั้งการกำหนดลำดับขั้นตอนในการทำงานเรียบร้อยแล้ว ขั้นตอนต่อไปจึงเป็นการสร้างโปรแกรมโดยใช้ข้อมูลดังกล่าว ประกอบในการเขียนโปรแกรมซึ่งโปรแกรมที่สร้างขึ้นก็คือ ชุดคำสั่งของ DBASE III - Plus เพื่อทำหน้าที่เรียกใช้ฐานข้อมูลที่เกี่ยวข้อง มาทำการแสดงข้อมูล หรือปรับปรุง แก้ไข ข้อมูล ตามแต่ละประเภท และลำดับในการทำงานสำหรับขั้นตอนการทำงานของแต่ละโปรแกรมได้อธิบายไว้ในรูปแบบของ Program Specification ซึ่งเป็นการกำหนดคุณสมบัติของแต่ละโปรแกรม นอกจากนี้ยังเป็นการอธิบาย Process การทำงานของโปรแกรมว่ามีขั้นตอนอย่างไร ซึ่งรูปแบบของการ



รูปที่ 6.1 แสดงลำดับการทำงานของโปรแกรมห้องสมุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายนี้อาจแสดงอยู่ในรูปแบบของภาษาเขียน หรือ Process Chart หรือ Psuedo Code สำหรับ Program Spec. ของระบบห้องสมุดนี้ได้ออกแบบและเขียนอยู่ในรูปของ Psuedo Code เพื่อเป็นการง่ายต่อการที่จะนำไปแปลงเป็นคำสั่งใน DBASE III - Plus

รายละเอียดของ Program Spec. และ Program Source Listing ดูได้จาก ภาคผนวก

#### 6.6 การสร้างและพัฒนาโปรแกรมแอสเซมบลี เพื่อใช้ในการอ่านข้อมูล

เนื่องจากโปรแกรมห้องสมุดนี้ได้ออกแบบให้มีการรับข้อมูลได้ทั้งจากคีย์บอร์ด และจาก เครื่องอ่านบาร์โคด ดังนั้นจึงต้องมีการสร้างโปรแกรมแอสเซมบลี เพื่อให้สามารถรับข้อมูลจาก พอร์ทอินพุตแบบอนุกรมของเครื่องไมโครคอมพิวเตอร์ เนื่องจากในภาษา DBASE III - Plus ไม่มีคำสั่งที่จะทำงานดังกล่าวข้างต้น ขณะเดียวกันก็ได้รับข้อมูลจาก Keyboard ด้วยโปรแกรม แอสเซมบลีที่สร้างขึ้นนี้เป็นภาษาแอสเซมบลีของ ซีพียู 8088 ซึ่งแบ่งออกเป็น 2 โปรแกรม คือ โปรแกรม ชื่อ "INIT 8250 .ASM" เป็นโปรแกรมที่ใช้สำหรับ Initialize ส่วน ฮาร์ดแวร์ ซึ่งทำหน้าที่เป็น Serial Communication Processor (8250) ให้สามารถรับ ข้อมูลจากเครื่องอ่านบาร์โคด

โปรแกรม ชื่อ "RECEIVE . ASM" เป็นโปรแกรมที่จะถูกเรียกใช้โดย โปรแกรม DBASE III - Plus ซึ่งทำหน้าที่ในการรับข้อมูลจากเครื่องอ่านบาร์โคด และ Keyboard โดย โปรแกรมแอสเซมบลีนี้ซึ่งจะถูกเรียกใช้โดย โปรแกรม DBASE III - Hus ได้ต้องอยู่ในรูปของ Binary File สำหรับการทำให้ Source Program ภาษา แอสเซมบลี ให้อยู่ในรูปของ Binary File มีขั้นตอนดังนี้

A > MASM RECEIVE RECEIVE

A > LINK RECEIVE RECEIVE

A > EXE 2 BIN RECEIVE

รายละเอียดของโปรแกรมแอสเซมบลี ดูได้ในภาคผนวก

### 6.7 รายละเอียดเกี่ยวกับโปรแกรมที่ใช้ในระบบห้องสมุด

สำหรับโปรแกรมทั้งหมดที่ใช้สำหรับในการทำงานในห้องสมุด ประกอบด้วยทั้งหมด 11 โปรแกรม โดยมีโปรแกรมที่เป็น DBASE III - Plus 9 โปรแกรม ซึ่งจะมีรายละเอียด Program Spec. และการแสดงทางจอภาพ (Screen Layout) ประกอบด้วย ส่วนอีก 2 โปรแกรมจะเป็นโปรแกรมภาษาแอสเซมบลี ซึ่งโปรแกรมทั้งหมดมีชื่อและการใช้งานดังต่อไปนี้

1. LIBRARY. PRG เป็นโปรแกรมสำหรับใช้เลือกรายการ การทำงานในห้องสมุด มีรายละเอียดของ Program Spec. ประกอบใน Document NO. P\_LIB00 และการแสดงจอภาพ (Screen Layout) Document No. S\_LIB00
2. SEARCH. PRG เป็นโปรแกรมสำหรับเลือกวิธีการค้นหาหนังสือ มี Program Spec. ประกอบใน Doc. No. P\_LIB01 และ Screen Layout ใน Doc. No. S\_LIB01
3. ACCESSION. PRG ใช้ในการค้นหาหนังสือด้วยเลขทะเบียนหนังสือ มี Program Spec ประกอบใน Doc. No. P\_LIB01\_1 และ Screen Layout ใน Doc. No. S\_LIB01\_1
4. CALL. PRG ใช้ในการค้นหาหนังสือด้วยเลขเรียกหนังสือ มี Pro. Spec ประกอบใน Doc. No. P\_LIB01-2 และ Scr. Lay. ใน Doc. No. S\_LIB01-2
5. AUTHOR. PRG ใช้ในการค้นหาหนังสือด้วยชื่อผู้แต่งมี Pro. Spec ประกอบใน Doc. No. P\_LIB01-3 และ Scr. Lay. ใน Doc. No. S\_LIB01-3
6. TITAL. PRG. ใช้ในการค้นหาหนังสือด้วยชื่อเรื่อง มี Pro. Spec ประกอบใน Doc. No. P\_LIB01-4 และ Scr. Lay. ใน Doc. No. S\_LIB01-4
7. BORROW. PRG เป็นโปรแกรมสำหรับใช้ในงานยืมหนังสือ มี Pro. Spec. ประกอบใน Doc. No. P\_LIB02 และ Scr. Lay. ใน Doc. S\_LIB02
8. RETURN. PRG เป็นโปรแกรมสำหรับใช้ในงานคืนหนังสือ มี Pro. Spec. ประกอบใน Doc. No. P-LIB03 และ Scr. Lay. ใน S\_LIB03

๑. DUEDATE. PRG เป็นโปรแกรมสำหรับใช้ในการตรวจหาหนังสือที่ครบกำหนดส่งคืน  
มี Pro. Spec. ใน Doc. No. P\_LIB04 และ Scr. Lay. ใน Doc. No.  
S\_LIB04

ส่วนอีก ๒ โปรแกรมที่เหลือเป็นโปรแกรมภาษาแอสเซมบลีมีดังนี้

1. INIT 8250. BIN เป็นโปรแกรมสำหรับใช้ในการ Initialize ตัว Serial  
Communication Processor (8250) ให้ทำงานในการรับข้อมูลจากเครื่องอ่าน  
บาร์โคดได้
2. RECEIVE. BIN เป็นโปรแกรมสำหรับใช้ในการรับข้อมูลจากเครื่องอ่านบาร์โคด  
หรือ คีย์บอร์ด จากนั้นจึงส่งข้อมูลที่ได้รับให้กับโปรแกรม DBASE III - Plus

บทที่ 7

บทสรุป

วิทยานิพนธ์นี้ เสนอผลงานวิจัย เกี่ยวกับการนำเอาบาร์โคดมาใช้ประโยชน์ในการทำงาน สำหรับห้องสมุด ร่วมกับการใช้โปรแกรมห้องสมุดซึ่งได้พัฒนาขึ้นสำหรับใช้กับไมโครคอมพิวเตอร์ เพื่อเป็นการอำนวยความสะดวกและเพิ่มประสิทธิภาพในการทำงานของเจ้าหน้าที่ อีกทั้งทำให้ขนาดของ บัตรห้องสมุดมีขนาดเล็กลง เพราะไม่ต้องใช้เนื้อที่ในการบันทึก และบัตรบันทึกประจำหนังสือก็ไม่จำเป็นต้องมีอีกต่อไป เป็นลดการสูญหายและความผิดพลาดในการบันทึก การทำงานของเครื่องอ่านบาร์-โคดและโปรแกรมนี้ได้ออกแบบให้ง่ายต่อการใช้งานโดยผู้ใช้ไม่จำเป็นต้องมีความรู้ทางด้านโปรแกรม มาก่อน สำหรับการใช้เครื่องอ่านบาร์โคดจากการทดลองพบว่า ผู้ใช้ส่วนใหญ่สามารถใช้หัวอ่านโดย เครื่องอ่านบาร์โคดสามารถอ่านได้อย่างถูกต้องตั้งแต่เริ่มใช้งาน ซึ่งผู้ใช้ในบางส่วนอาจต้องทดลอง เพื่อหาความเหมาะสมในการลากหัวอ่านโดยประมาณ 2-3 ครั้ง จึงจะสามารถใช้งานได้ถูกต้อง และ เมื่อผู้ใช้มีความคุ้นเคยกับการใช้งานแล้วก็สามารถใช้เครื่องอ่านบาร์โคดได้อย่างถูกต้องตลอดไป สำหรับความผิดพลาดที่อาจเกิดขึ้นในบางครั้งทำให้เครื่องอ่านไม่ได้ โดยส่วนใหญ่มีสาเหตุมาจาก ความร้อนของผู้ใช้ ซึ่งอาจเกิดจากการจับหัวอ่านไม่อยู่ในระนาบเสมอกันกับบาร์โคดในขณะที่ทำการ ลากหัวอ่าน ซึ่งทำให้สัญญาณที่ได้จากการสะท้อนของอินฟราเรดบนบาร์โคดคลาดเคลื่อนไป และบาง ครั้งอาจเกิดจากการลากด้วยความเร็วที่ไม่สม่ำเสมอ ซึ่งมีสาเหตุมาจากการลากด้วยความเร็วที่ช้าเกินไป ในการแก้ไขผู้ใช้ควรจับหัวอ่านให้อยู่ในระนาบเดียวกับบาร์โคด และลากผ่านไปบนบาร์โคด ให้เร็วขึ้น สำหรับการส่งข้อมูลของเครื่องอ่านบาร์โคดที่อ่านได้ไปยังไมโครคอมพิวเตอร์ พบว่าข้อมูล ที่ปรากฏบนเครื่องไมโครคอมพิวเตอร์ตรงกับข้อมูลของบาร์โคดทุกครั้ง

แนวความคิดและข้อเสนอแนะ เพื่อเป็นแนวทางในการพัฒนาเครื่องอ่านบาร์โคดให้มี ประสิทธิภาพในการทำงานยิ่งขึ้นต่อไป มีดังนี้

1. เครื่องอ่านบาร์โคดที่ได้ออกแบบสร้างขึ้นอาจจะมีความที่ค่อนข้างใหญ่ เนื่องจาก ระบบไมโครโปรเซสเซอร์ ซีพียู Z-80 ต้องประกอบด้วยอุปกรณ์มากขึ้นซึ่งในการ

พัฒนาต่อไปอาจใช้อุปกรณ์จำพวก Single Chip Microcomputer เช่นตระกูลของ 48-Series ก็อาจจะทำให้ขนาดของเครื่องเล็กลงได้

2. ในการใช้งานจริงข้อมูลต่าง ๆ ของห้องสมุดมีเป็นจำนวนมากและมีแต่จะเพิ่มขึ้นเรื่อย ๆ ดังนั้นการใช้ Hard Disk สำหรับไมโครคอมพิวเตอร์ในเก็บชั้นตอนต่าง ๆ ในการทำโปรแกรม และเก็บข้อมูลของห้องสมุด จะเป็นวิธีที่ทำให้การทำงานรวดเร็วยิ่งขึ้นและสามารถเก็บข้อมูลได้มากขึ้น เนื่องจาก Floppy Disk ธรรมดามีความจุข้อมูลเพียง 360 KBYTE เทียบกับความจุของ Hard Disk ซึ่งมีขนาดตั้งแต่ 10 MBYTE ขึ้นไปจะเห็นได้ว่าแตกต่างกันอย่างมาก นอกจากนี้การใช้เครื่องไมโครคอมพิวเตอร์ที่มีความเร็วในการทำงานสูงขึ้น ก็จะมีส่วนช่วยให้การทำงานของโปรแกรมรวดเร็วขึ้น
3. สำหรับการสร้างบาร์โคดขึ้นใช้งานในวิทยานิพนธ์นี้ ในตอนแรกได้ใช้วิธีการย่อขนาดจากบาร์โคดต้นแบบด้วยการถ่ายเอกสารย่อ ซึ่งพบว่าเมื่อย่อขนาดลงมา ลักษณะของบาร์โค้ดที่ได้ผิดไปจากเดิม คือมีลักษณะคอคดลง ทำให้สัดส่วนของบาร์โคดที่ได้จากวิธีนี้คลาดเคลื่อน และไม่สามารถนำไปใช้งานได้ ดังนั้นจึงได้เปลี่ยนเป็นวิธีการใช้ถ่านฟิล์มย่อลงมา ซึ่งบาร์โคดที่ได้มีสัดส่วนที่ถูกต้องเหมือนต้นแบบและสามารถใช้งานได้ดี แต่มีข้อเสียคือค่าใช้จ่ายในการทำค่อนข้างสูง ซึ่งอาจไม่เหมาะสมกับการนำไปใช้ในทางปฏิบัติดังนั้นจึงควรหาวิธีอื่นในการสร้างที่เหมาะสมกับการใช้งานและมีคุณภาพที่ดีต่อไป

สำหรับประโยชน์ของการใช้บาร์โค้ดนี้ ยังสามารถที่จะนำไปพัฒนา เพื่อใช้ในระบบงานอื่น ๆ ได้อีกมาก เช่นในการลงทะเบียนนักศึกษา การลงหน่วยกิตวิชาเรียน เป็นต้น แล้วแต่ความเหมาะสมในการนำไปตัดแปลงใช้งาน รวมทั้งการพัฒนาไปสู่ระบบออนไลน์ (Online Network) เพื่อเชื่อมโยงการทำงานของทุกระบบเข้าด้วยกัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ รองศาสตราจารย์ ดร.สิทธิชัย โภไคยอุดม ซึ่งเป็นอาจารย์ที่ปรึกษา  
ได้กรุณาให้คำแนะนำจนวิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี ขอขอบคุณ อาจารย์ พลผดุง ผดุงกุล  
และคุณไพศาล อัครวิมล ที่ให้คำแนะนำและสนับสนุนมาโดยตลอด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. Handbook, " Z80 - CPU Technical Manual.", Zilog Inc., U.S.A.
2. Handbook, " Z80 - P10 Technical Manual.", Zilog Inc., U.S.A.
3. Handbook, " Z80 - CTC Technical Manual.", Zilog Inc., U.S.A.
4. "M 6800 Application Manual.", Motorola Semiconductor Products Inc., 1975
5. Handbook, "The IBM Personal Computer Technical Reference Manual.
6. Handbook, "The IBM Personal Computer Disk Operating System Manual."
7. Handbook, "The IBM Personal Computer Macro Assembler Manual."
8. David C. Willen and Jeffrey I. Kranta., "8088 Assembler Language Programming The IBM PC.," Howard W. Sams & Co., Inc.
9. George Tsu-der Chou., "DBASE III PLUS Handbook.", Que Corporation
10. ยืน ภู่วรรณ , ชัยยงค์ วงศ์ชัยสุวัฒน์ , ไพศาล สงวนหนู "เทคโนโลยี ไมโครคอมพิวเตอร์ 16 บิต " , บริษัทซีเอ็ดยูเคชั่นจำกัด
11. วุฒิชัย รุจิระประภา, " การออกแบบและพัฒนาระบบออนไลน์ NEC ACOS-4", วิทยานิพนธ์ สำหรับปริญญาวิศวกรรมศาสตรมหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า ลาดกระบัง พ.ศ. 2525

ภาคผนวกที่ 1

แสดงโปรแกรมของระบบไมโครโปรเซสเซอร์ที่ใช้ในเครื่องอ่านบาร์โค้ด



0000 0000 DIGITAL WAND CONTROL PROGRAM

0605
0607
1609
160A
160C
1690
1692
169E
169F
16A0
16A2
16A4
16A6
16AB
16AA
16AC
16AE
16B0
16B2
16B4
06B6
06C0
16CA
06D0

STCONT EQU 0605H
CTC1NT EQU 0607H
BARCNT EQU 1609H
TMPTR EQU 160AH
TMBUF EQU 160CH
UPCPTR EQU 1690H
UPCBUF EQU 1692H
CHRCNT EQU 169EH
RCHAR EQU 169FH
TM EQU 16A0H
.75TM EQU 16A2H
SAMPT EQU 16A4H
BEND1 EQU 16A6H
BEND2 EQU 16ABH
BEND3 EQU 16AAH
BEND4 EQU 16ACH
BAR1 EQU 16AEH
BAR2 EQU 16B0H
BAR3 EQU 16B2H
BAR4 EQU 16B4H
ASCTAB EQU 06B6H
UPCTAB EQU 06C0H
OUTBUF EQU 16CAH
SERNIT EQU 06D0H

ORG 0
; THIS ROUTINE INITIALIZE THE STACK POINTER
; 8251 USART, Z-80 CTC, Z-80 PID AND LIGHTS
; THE 'GOOD' LED AT THE FRONT PANAL.

0000 31 1800
0003 21 06D0
0006 7E
0007 FE FF
0009 28 05
000B D3 C1
000D 23
000E 18 F6
0010 21 0605
0013 06 02
0015 0E 40
0017 ED B3
0019 21 0607
001C 06 02
001E 0E 41
0020 ED B3
0022 3E 0F
0024 D3 B3
0026 3E FF
0028 D3 B1
002A 0E 02

LD SP,1800H
HL,SERNIT
SERCON: LD A,(HL)
CP OFFH
JR Z,SEREND
OUT (0C1H),A
HL SERCON
SEREND: LD HL,CTCONT
LD B,2
LD C,40H
OTIR
LD HL,CTC1NT
LD B,2
LD C,41H
OTIR
LD A,0FH
OUT (83H),A
GOOD: LD A,-1
OUT (81H),A
LD C,2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

002C' CD 0095' ..L1: CALL RSCTC
002F' CD 00AC' ..L2: CALL RCCTC
0032' 21 E000 LD HL,0E000H
0035' AF XOR A
0036' ED 52 SBC HL,DE
003U' 30 EF JR NC,..L2
003A' 0D DEC C
003B' 20 EF JR NZ,..L1
003D' AF XOR A
003E' D3 81 OUT (B1H),A

```

```

; THIS ROUTINE READ AND STORE THE TIME
; REQUIRED FOR READING EACH BAR FOR LATER
; USE. ALSO VERIFY THAT NO BAR IS WIDER THAN
; THE LIMITATION, WHICH IS DERIVED FROM THE
; SLOWEST SCAN RATE OF 0.5 INCH/SECOND PER
; STANDARD UPC MODULE WIDTH.
;
;

```

```

0040' YKWAND: LD A,17
0040' 3E 11 ERR: LD (BARCNT),A
0042' 32 1609 LD HL,TMBUF
0045' 21 160C LD (TMPTR),HL
0048' 22 160A START: IN A,(BOH)
004B' DB 80 AND 1
004D' E6 01 JR NZ,START
004F' 20 FA ..1STBR: IN A,(BOH)
0051' DB 80 AND 1
0053' E6 01 JR Z,..1STBR
0055' 28 FA ..BAR: CALL RSCTC
0057' CD 0095' ..RBR: CALL RCCTC
005A' CD 00AC' IN A,(BOH)
005D' DB 80 AND 1
005F' E6 01 JR Z,..SP
0061' 28 0A LD HL,3250*4
0063' 21 32CB OR A
0066' B7 SBC HL,DE
0067' ED 52 JR NC,..RBR
0069' 30 EF ..JST: JR START
006B' 18 DE ..SP: CALL TMSAV
006D' CD 00B9' CALL RSCTC
0070' CD 0095' LD A,(BARCNT)
0073' 3A 1609 DEC A
0076' 3D JP Z,WSORT
0077' CA 00C4' LD (BARCNT),A
007A' 32 1609 ..RSP: CALL RCCTC
007D' CD 00AC' IN A,(BOH)
0080' DB 80 AND 1
0082' E6 01 JR NZ,..BR
0084' 20 0A LD HL,3250*4
0086' 21 32CB OR A
0089' B7 SBC HL,DE
008A' ED 52 JR NC,..RSP
008B' 30 EF ..JST
008E' 18 DB ..BR: CALL TMSAV
0090' CD 00B9' JR ..BAR
0093' 18 C2

```

```

0095' 21 0605      ;
0098' 7E          RSCCTC: LD      HL,CTCONT
0099' F6 02      LD      A,(HL)
009B' D3 40      OR      2
009D' 23        OUT     (40H),A
009E' 7E        INC     HL
009F' D3 40      LD      A,(HL)
00A1' 23        OUT     (40H),A
00A2' 7E        INC     HL
00A3' F6 02      LD      A,(HL)
00A5' D3 41      OR      2
00A7' 23        OUT     (41H),A
00A8' 7E        INC     HL
00A9' D3 41      LD      A,(HL)
00AB' C9        OUT     (41H),A
                RET

```

```

00AC' DB 40      ;
00AE' 47        RCCTC: IN      A,(40H)
00AF' AF        LD      B,A
00B0' 90        XOR     A
00B1' 5F        SUB     B
00B2' DB 41      LD      E,A
00B4' 47        IN      A,(41H)
00B5' AF        LD      B,A
00B6' 90        XOR     A
00B7' 57        SUB     B
00B8' C9        LD      D,A
                RET

```

```

00B9' 2A 160A    ;
00BC' 73        TMSAV: LD      HL,(TMPTR)
00BD' 23        LD      (HL),E
00BE' 72        INC     HL
00BF' 23        LD      (HL),D
00C0' 22 160A   INC     HL
00C3' C9        LD      (TMPTR),HL
                RET

```

```

; THIS ROUTINE CONVERTS THE TIME SLOTS READ BY
; THE YKWAND ROUTINE TO UPC BINARY NUMBER.
; ALSO CHECK IF ANY GROUP OF BAR, WHICH WILL
; PRODUCE A UPC NUMBER, IS NOT RELIABLE ENOUGH
; TO BE PROCESSED.

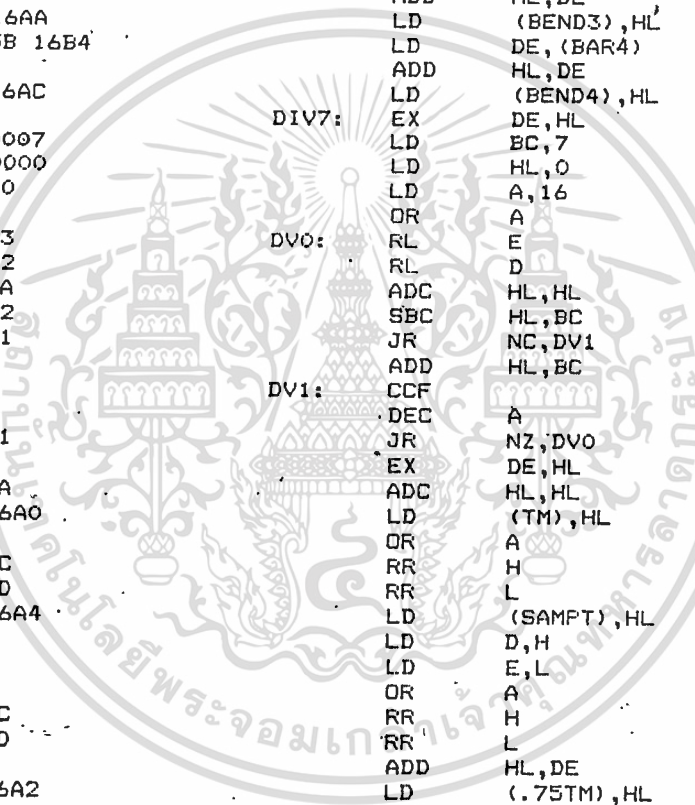
```

```

00C4' 3E 06      ;
00C6' 32 169E   WSORT: LD      A,6
00C9' 21 1692   LD      (CHRCNT),A
00CC' 22 1690   LD      HL,UPCBUF
00CF' 21 160C   LD      (UPCPTR),HL
00D2' 22 160A   LD      HL,TMBUF
00D5' 11 0006   LD      (TMPTR),HL
00DB' 19        SKPGRD: LD      DE,6
00D9' 22 160A   ADD     HL,DE
00DC' 2A 160A   LD      (TMPTR),HL
00DF' 01 000B   ..WSRT: LD      HL,(TMPTR)
00E2' 11 16AE   LD      BC,8
00E5' ED B0     LD      DE,BAR1
                LDIR

```

00E7	22 160A	LD	(TMFTR),HL
00EA	2A 16AE	LD	HL,(BAR1)
00ED	22 16A6	LD	(BEND1),HL
00F0	ED 5B 16B0	LD	DE,(BAR2)
00F4	19	ADD	HL,DE
00F5	22 16AB	LD	(BEND2),HL
00FB	ED 5B 16B2	LD	DE,(BAR3)
00FC	19	ADD	HL,DE
00FD	22 16AA	LD	(BEND3),HL
0100	ED 5B 16B4	LD	DE,(BAR4)
0104	19	ADD	HL,DE
0105	22 16AC	LD	(BEND4),HL
0108	EB	DIV7:	EX DE,HL
0109	01 0007	LD	BC,7
010C	21 0000	LD	HL,0
010F	3E 10	LD	A,16
0111	B7	OR	A
0112	CB 13	DVO:	RL E
0114	CB 12	RL	D
0116	ED 6A	ADC	HL,HL
0118	ED 42	SBC	HL,BC
011A	30 01	JR	NC,DV1
011C	09	ADD	HL,BC
011D	3F	DV1:	CCF
011E	3D	DEC	A
011F	20 F1	JR	NZ,DVO
0121	EB	EX	DE,HL
0122	ED 6A	ADC	HL,HL
0124	22 16A0	LD	(TM),HL
0127	B7	OR	A
0128	CB 1C	RR	H
012A	CB 1D	RR	L
012C	22 16A4	LD	(SAMPT),HL
012F	54	LD	D,H
0130	SD	LD	E,L
0131	B7	OR	A
0132	CB 1C	RR	H
0134	CB 1D	RR	L
0136	19	ADD	HL,DE
0137	22 16A2	LD	(.75TM),HL
013A	EB	EX	DE,HL
013B	3E 04	LD	A,4
013D	01 16AE	LD	BC,BAR1
0140	08	CPRE:	EX AF,AF
0141	0A	LD	A,(BC)
0142	6F	LD	L,A
0143	03	INC	BC
0144	0A	LD	A,(BC)
0145	67	LD	H,A
0146	B7	OR	A
0147	ED 52	SBC	HL,DE
0149	30 0A	JR	NC,BAR0K
014B	2A 16A0	LD	HL,(TM)
014E	7C	LD	A,H
014F	02	LD	(BC),A
0150	0B	DEC	BC



```

0151' 7D LD A,L
0152' 02 LD (BC),A
0153' 18 75 JR ..WS1
0155' 03 BAROK: INC BC
0156' 08 EX AF,AF'
0157' 3D DEC A
0158' 20 E4 JR NZ,..CPRE
015A' 06 07 RECOG: LD B,7
015C' 0E 00 LD C,0
015E' 18 01 .MOD1: JR .SHO
0160' 37 .SH1: SCF
0161' CB 11 .SHO: RL C
0163' 05 DEC B
0164' 2A 16A0 LD HL,(TM)
0167' ED 5B 16A4 LD DE,(SAMPT)
016B' 19 ADD HL,DE
016C' 22 16A4 LD (SAMPT),HL
016F' EB EX DE,HL
0170' 2A 16A6 LD HL,(BEND1)
0173' B7 OR A
0174' ED 52 SBC HL,DE
0176' 30 E6 JR NC,..MOD1
0178' 2A 16AB LD HL,(BEND2)
017B' B7 OR A
017C' ED 52 SBC HL,DE
017E' 38 03 JR C,..MOD3
0180' B7 .SH: OR A
0181' 18 DD JR .SH1
0183' 2A 16AA .MOD3: LD HL,(BEND3)
0186' B7 OR A
0187' ED 52 SBC HL,DE
0189' 30 D3 JR NC,..MOD1
018B' 2A 16AC LD HL,(BEND4)
018E' B7 OR A
018F' ED 52 SBC HL,DE
0191' 30 ED JR NC,..SH
0193' 2A 1690 LD HL,(UPCPTR)
0196' 71 LD (HL),C
0197' 23 INC HL
0198' 22 1690 LD (UPCPTR),HL
019B' 3E FF LD A,-1
019D' 77 LD (HL),A
019E' 21 169E LD HL,CHRCNT
01A1' 35 DEC (HL)
01A2' C2 00DC JP NZ,..WSRT
01A5' C3 01AB JP WCNVRT

```

```

;
; THIS ROUTINE CONVERTS THE UPC BINARY NUMBER
; PRODUCED BY WSORT TO ASCII NUMBER.
;

```

```

01AB' 11 1692 WCNVRT: LD DE,UPCBUF
01AB' 21 16CA LD HL,OUTBUF
01AE' 01 000A LD BC,10
01B1' E5 PUSH HL
01B2' 21 06C0 LD HL,UPCTAB
01B5' 0E 0A LD C,10

```

```

01B7' 1A      ..RD: LD A,(DE)
01B8' FE FF   CP -1
01BA' 20 05   JR NZ,..NFF
01BC' E1      POP HL
01BD' 77      LD (HL),A
01BE' C3 01DE' JP LIGHT
01C1' BE      ..NFF: CP (HL)
01C2' 28 0A   JR Z,MATCH
01C4' 0D      DEC C
01C5' 20 04   JR NZ,..OK
01C7' E1      POP HL
01C8' C3 0040' JP ERR
01CB' 23      ..OK: INC HL
01CC' 18 F3   JR ..NFF
01CE' E5      MATCH: PUSH HL
01CF' 21 06B6 LD HL,ASCTAB
01D2' 09      ADD HL,BC
01D3' 7E      LD A,(HL)
01D4' E1      POP HL
01D5' D9      EXX
01D6' E1      POP HL
01D7' 77      LD (HL),A
01D8' 23      INC HL
01D9' E5      PUSH HL
01DA' D9      EXX
01DB' 13      INC DE
01DC' 18 D9   JR ..RD

```

; THIS ROUTINE LIGHTS THE 'GOOD' LED AND  
; TRANSMIT THE ASCII NUMBERS VIA 8251 TO  
; THE HOST COMPUTER.

```

01DE' 3E FF   LIGHT: LD A,-1
01E0' D3 81   OUT (81H),A
01E2' 06 06   RS: LD B,6
01E4' 21 16CA ..STS: LD HL,OUTBUF
01E7' DB C1   ..STS: IN A,(0C1H)
01E9' E6 01   AND 1
01EB' 28 FA   JR Z,..STS
01ED' 7E      LD A,(HL)
01EE' FE FF   CP -1
01F0' CA 0026' JP Z,GOOD
01F3' D3 C0   OUT (0C0H),A
01F5' 23      INC HL
01F6' 18 EF   JR ..STS

```

END

DIGITAL WAND CONTROL PROGRAM

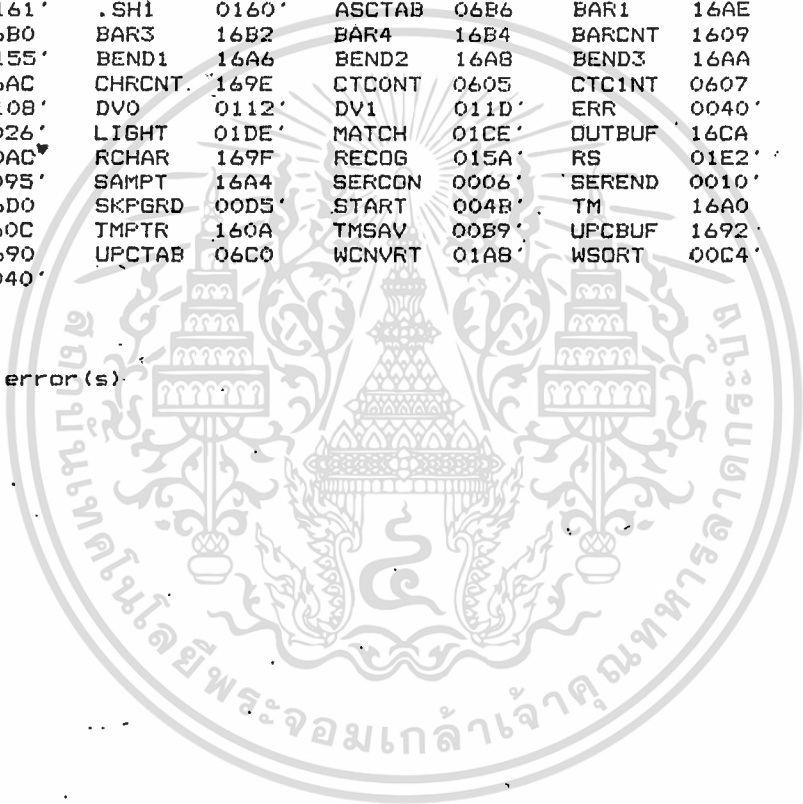
PAGE 5

Macros:

Symbols:

..BAR	0057'	..BR	0090'	..JST	006B'	..L1	002C'
..L2	002F'	..NFF	01C1'	..OK	01CB'	..RBAR	005A'
..RD	01B7'	..RSP	007D'	..SH	0180'	..SP	006D'
..STS	01E7'	..WS1	00EA'	..WSRT	00DC'	..1STBR	0051'
.7STM	16A2	.CPRE	0140'	.MOD1	015E'	.MOD3	0183'
.SHO	0161'	.SH1	0160'	ASCTAB	06B6	BAR1	16AE
BAR2	16B0	BAR3	16B2	BAR4	16B4	BARCNT	1609
BAROK	0155'	BEND1	16A6	BEND2	16A8	BEND3	16AA
BEND4	16AC	CHRCNT	169E	CTCONT	0605	CTCINT	0607
DIV7	0108'	DVO	0112'	DV1	011D'	ERR	0040'
GOOD	0026'	LIGHT	01DE'	MATCH	01CE'	OUTBUF	16CA
RCCTC	00AC'	RCHAR	167F	RECOG	015A'	RS	01E2'
RSCTC	0095'	SAMPT	16A4	SERCON	0006'	SEREND	0010'
SERNIT	06D0	SKPGRD	00D5'	START	0048'	TM	16A0
TMBUF	160C	TMPTR	160A	TMSAV	00B9'	UPCBUF	1692
UPCPTR	1690	UPCTAB	06C0	WCNVRT	01AB	WSORT	00C4'
YKWAND	0040'						

No Fatal error(s)



A>

ภาคผนวกที่ 2

แสดง File Layout ของฐานข้อมูลที่ใช้ในโปรแกรมห้องสมุด



FILE LAYOUT

FILE NAME : BOOKS.DBF

DESCRIPTION : BOOKS file consist of Books detail, Reserver\_ID : Status of the book, Expire date, and Borrower\_ID.

RECORD LENGTH : 216 Byte

FILE ORGANIZATION : Database

DEVICE : Floppy disk

RECORD KEY : ACCS\_NO, CALL\_NO, AUTHOR, TITLE, EXPIRE

[ RECORD DESCRIPTION ]

Ref. No.	Variable name	Length	Type	Position	Remark
001	ACCS_NO	06	Charactr	1- 6	[Key]
002	CALL-NO	15	Charactr	7- 21	[Key]
003	AUTHOR	50	Charactr	22- 71	[Key]
004	TITLE	100	Charactr	72-171	[Key]
005	RESV_ID1	06	Charactr	172-177	
006	RESV_ID2	06	Charactr	178-183	
007	RESV_ID3	06	Charactr	184-189	
008	RESV_ID4	06	Charactr	190-195	
009	RESV_ID5	06	Charactr	196-201	
010	BOOK_STAT	01	Numeric	202-202	0 - In Library 1 - Borrowed 2 - Reserved
011	EXPIRE	08	Date	203-210	[Key]
012	BRRW_ID	06	Charactr	211-216	



ภาคผนวกที่ 3

แสดง Screen Layout ของโปรแกรมห้องสมุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Document No: S\_LIB00

SCREEN LAYOUT

DATE: mm/dd/yy

KMIT LADKRABANG LIBRARY

LIBRARY SYSTEM

1. SEARCH THE BOOK
2. BORROW THE BOOK
3. RETURN THE BOOK
4. BORROWER DUE DATE
- X. EXIT THE SYSTEM

Please enter your selection: [ MESSAGE DISPLAY LINE ]

Please wait...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KMIT LADKRABANG LIBRARY

DATE: mm/dd/yy

SEARCH THE BOOK

1. ACCESSION NUMBER

2. CALL NUMBER

3. AUTHOR

4. TITLE

E. END



Please enter your selection:

[ MESSAGE DISPLAY LINE ]

Please wait...

-----  
SCREEN LAYOUT  
-----

KMIT LADKRABANG LIBRARY

DATE: ##/dd/yy

SEARCHING KEY IS --

SEARCH THE BOOK

ACCESSION NUMBER: xxxxxx

CALL NUMBER: -----

AUTHOR: -----

TITLE: -----

( MESSAGE DISPLAY LINE )

ID: -----

STATUS: -----

NAME: -----

USER\_DEPT: -----

( MESSAGE DISPLAY LINE )

Please wait....

Document No.: S\_LIB01\_2

-----  
SCREEN LAYOUT  
-----

DATE: ##/dd/yy

SEARCHING KEY IS --

SEARCH THE BOOK

CALL NUMBER: xxxxxx

ACCESSION NUMBER: xxxxxx

CALL NUMBER: -----

AUTHOR: -----

TITLE: -----

( MESSAGE DISPLAY LINE )

ID: -----

STATUS: -----

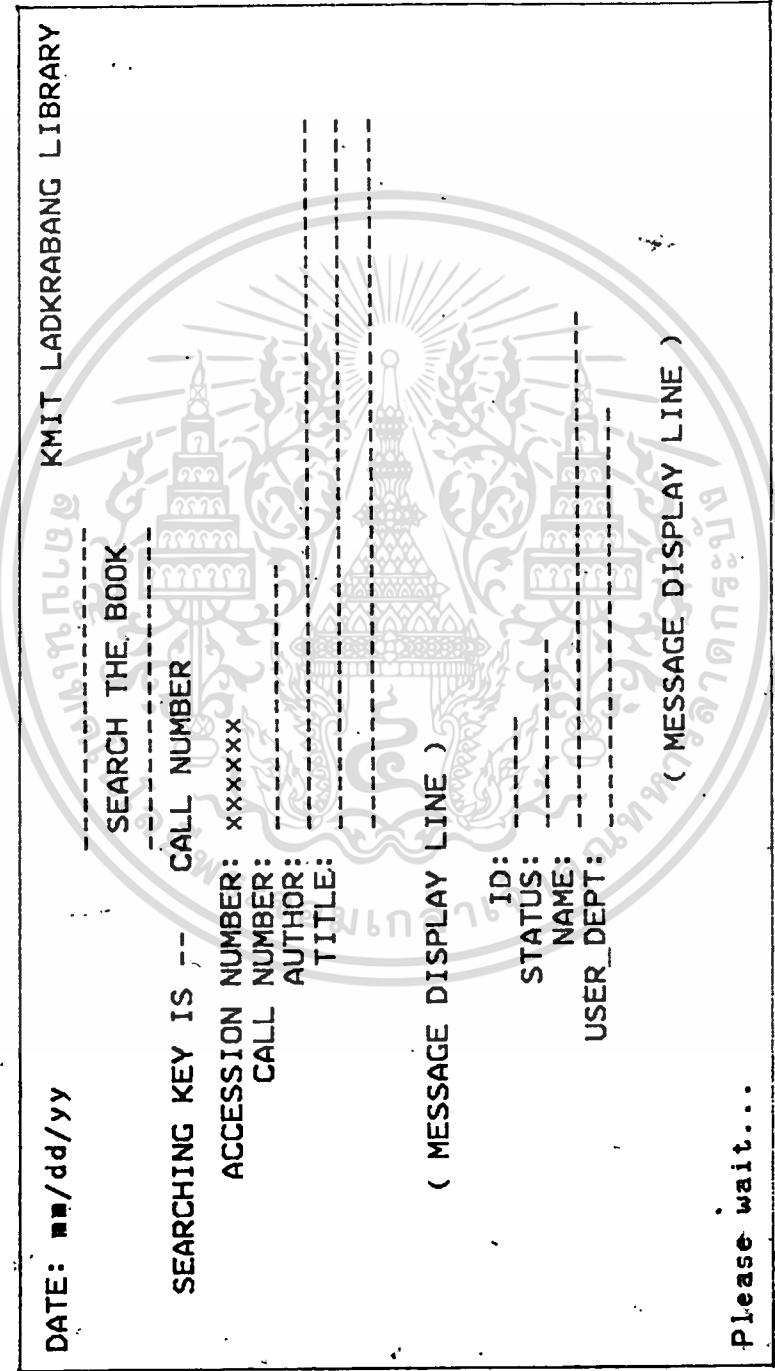
NAME: -----

USER\_DEPT: -----

( MESSAGE DISPLAY LINE )

Please wait....

KMIT LADKRABANG LIBRARY



Document No: S\_LIB01\_3

-----  
SCREEN LAYOUT  
-----

KMIT LADKRABANG LIBRARY

SEARCH THE BOOK

SEARCHING KEY IS -- AUTHOR

ACCESSION NUMBER: xxxxxx  
CALL NUMBER: -----  
AUTHOR: -----  
TITLE: -----

( MESSAGE DISPLAY LINE )

ID: -----  
STATUS: -----  
NAME: -----  
USER\_DEPT: -----

( MESSAGE DISPLAY LINE )

Please wait...

Document No: S\_LIB01\_4

SCREEN LAYOUT

KMIT LADKRABANG LIBRARY

SEARCH THE BOOK

DATE: mm/dd/yy

SEARCHING KEY IS -- TITLE

ACCESSION NUMBER: xxxxxx

CALL NUMBER:

AUTHOR:

TITLE:

( MESSAGE DISPLAY LINE )

ID:

STATUS:

NAME:

USER\_DEPT:

( MESSAGE DISPLAY LINE )

Please wait...

Document No: S\_LIB02

SCREEN LAYOUT

```

*-----*
* KMIT LADKRABANG LIBRARY *
*-----*
DATE: mm/dd/yy
Borrower:
ID: xxxxxx
STATUS: That is OK ? (Y/N): x
NAME:
DEPARTMENT:
MESSAGE DISPLAY LINE ]
[
BOOKS DETAIL
ACCESSION NUMBER: xxxxxx
CALL NUMBER:
AUTHOR:
TITLE:
[ MESSAGE DISPLAY LINE ]
Please wait...
*-----*

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Document No: S\_LIB03

SCREEN LAYOUT

KMIT LADKRABANG LIBRARY

DATE: mm/dd/yy

RETURN THE BOOK

BORROWER

ID: xxxxxx That is OK ? (Y/N): x

STATUS:

NAME:

DEPARTMENT:

MESSAGE DISPLAY LINE ]

BOOKS DETAIL

ACCESSION NUMBER: xxxxxx

CALL NUMBER:

AUTHOR:

TITLE:

That is OK ? (Y/N): x

[ MESSAGE DISPLAY LINE ]

Please wait...

Document No: S\_LIB04

SCREEN LAYOUT

KMIT LADKRABANG LIBRARY

DATE: mm/dd/yy.

BORROWER DUE DATE

Would you like to view the DUE DATE BOOK for today ... (Y/N):

BORROWER

ID:

STATUS:

NAME:

DEPARTMENT:

BOOKS DETAIL

ACCESSION NUMBER:

CALL NUMBER:

AUTHOR:

TITLE:

[ MESSAGE DISPLAY LINE ]

Please wait...

ภาคผนวกที่ 4

แสดง Program Specification ของโปรแกรมห้องสมุด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Document No: P\_LIB00

-----\*  
PROGRAM SPECIFICATION!  
-----\*

Task Name : Library System  
Program Name : LIB00  
Program Description : Library system main program

Screen : S\_LIB00

[ Process Description ]

```

CLEAR SCREEN
SET EXIT TO "N"
DO WHILE NOT EXIT
  DISPLAY SCREEN "S_LIB00"
  READ SELECTION
  CASE
    WHEN SELECTION = "1"
      PERFORM "SEARCH THE BOOK"
    WHEN SELECTION = "2"
      PERFORM "BORROW THE BOOK"
    WHEN SELECTION = "3"
      PERFORM "RETURN THE BOOK"
    WHEN SELECTION = "4"
      PERFORM "BORROWER DUE DATE"
    WHEN SELECTION = "5"
      PERFORM "RESERVE BOOK"
    WHEN SELECTION = "X"
      SET EXIT TO "Y"
      RETURN TO SYSTEM
    OTHERWISE
      WRITE "INVALID ENTRY PLEASE TRY AGAIN"
  ENDCASE
ENDDO

```

\*-----\*  
PROGRAM SPECIFICATION:  
\*-----\*

Task Name : SEARCH THE BOOK  
Program Name : LIB01  
Program Description : Search the book main program

Screen : S\_LIB01

[ Process Description ]

```
CLEAR SCREEN
SET EXIT TO "N"
DO WHILE NOT EXIT
  DISPLAY SCREEN "S_LIB01"
  READ SELECTION
  CASE
    WHEN SELECTION = "1"
      PERFORM "SEARCH BY ACCESSION"
    WHEN SELECTION = "2"
      PERFORM "SEARCH BY CALL_NO"
    WHEN SELECTION = "3"
      PERFORM "SEARCH BY AUTHOR"
    WHEN SELECTION = "4"
      PERFORM "SEARCH BY TITLE"
    WHEN SELECTION = "X"
      SET EXIT TO "Y"
      RETURN TO PREVIOUS PROGRAM
    OTHERWISE
      WRITE "INVALID ENTRY PLEASE TRY AGAIN"
  ENDCASE
ENDDO
```

Document No: P\_LIB01\_1

\*-----\*  
PROGRAM SPECIFICATION  
\*-----\*

Task Name : SEARCH BY ACCESSION\_NO  
Program Name : LIB01\_1  
Program Description : Program search the book by Accession\_No  
Input File : BOOK.DBF, USER.DBF  
Screen : S\_LIB01\_1

( Process Description )

CLEAR SCREEN  
ACCESS BOOKS FILE  
SET EXIT TO "N"  
DO WHILE NOT EXIT  
  DISPLAY SCREEN "S\_LIB01\_1" HEADER  
  SET ACCESSION\_NO TO 6 BLANK  
  READ ACCESSION\_NO  
  IF ACCESSION\_NO EQUAL 6 BLANK  
    CLEAR SCREEN  
    CLOSE BOOKS FILE  
    RETURN TO PREVIOUS PROGRAM  
  ELSE  
    VERIFY ACCESSION\_NO  
    IF FOUND THEN  
      RETRIVE BOOK DATA  
      DISPLAY INFORMATION  
    ELSE  
      WRITE "THIS BOOK IS BORROWED by:"  
      ACCESS USER FILE  
      USE BW\_ID TO VERIFY USER\_ID  
      IF FOUND THEN  
        RETRIVE USER DATA  
        DISPLAY INFORMATION  
      ENDIF  
    ENDIF  
  ENDIF  
ENDDO

Document No: P\_LIB01\_2

\*-----\*  
PROGRAM SPECIFICATION  
\*-----\*

Task Name : SEARCH BY CALL\_NO  
Program Name : LIB01\_2  
Program Description : Program search the book by Call\_No  
Input File : BOOKS.DBF, USER.DBF  
Screen : S\_LIB01\_2

( Process Description )

```
CLEAR SCREEN
ACCESS BOOKS FILE
SET EXIT TO "N"
DO WHILE NOT EXIT
  DISPLAY SCREEN "S_LIB01_2" HEADER
  SET CALL_NO TO 6 BLANK
  READ CALL_NO
  IF CALL_NO EQUAL 6 BLANK
    CLEAR SCREEN
    CLOSE BOOKS FILE
    RETURN TO PREVIOUS PROGRAM
  ELSE
    VERIFY CALL_NO
    IF FOUND THEN
      RETRIVE BOOK DATA
      DISPLAY INFORMATION
    ELSE
      WRITE "THIS BOOK IS BORROWED by:"
      ACCESS USER FILE
      USE BW_ID TO VERIFY USER_ID
      IF FOUND THEN
        RETRIVE USER DATA
        DISPLAY INFORMATION
      ENDIF
    ENDIF
  ENDIF
ENDDO
```

Document No: P\_LIB01\_3

\*-----\*  
PROGRAM SPECIFICATION  
\*-----\*

Task Name : SEARCH BY AUTHOR  
Program Name : LIB01\_3  
Program Description : Program search the book by AUTHOR NAME  
Input File : BOOKS.DBF, USER.DBF  
Screen : S\_LIB01\_3

( Process Description )

```
CLEAR SCREEN
ACCESS BOOKS FILE
SET EXIT TO "N"
DO WHILE NOT EXIT
  DISPLAY SCREEN "S_LIB01_3" HEADER
  SET AUTHOR TO 6 BLANK
  READ AUTHOR
  IF AUTHOR EQUAL 6 BLANK
    CLEAR SCREEN
    CLOSE BOOKS FILE
    RETURN TO PREVIOUS PROGRAM
  ELSE
    VERIFY AUTHOR
    IF FOUND THEN
      RETRIVE BOOK DATA
      DISPLAY INFORMATION
    ELSE
      WRITE "THIS BOOK IS BORROWED by:"
      ACCESS USER FILE
      USE BW_ID TO VERIFY USER_ID
      IF FOUND THEN
        RETRIVE USER DATA
        DISPLAY INFORMATION
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDDO
```

-----\*  
PROGRAM SPECIFICATION  
\*-----\*

Task Name : SEARCH BY TITLE  
Program Name : LIB01\_4  
Program Description : Program search the book by TITLE  
Input File : BOOKS.DBF, USER.DBF  
Screen : S\_LIB01\_4

( Process Description )

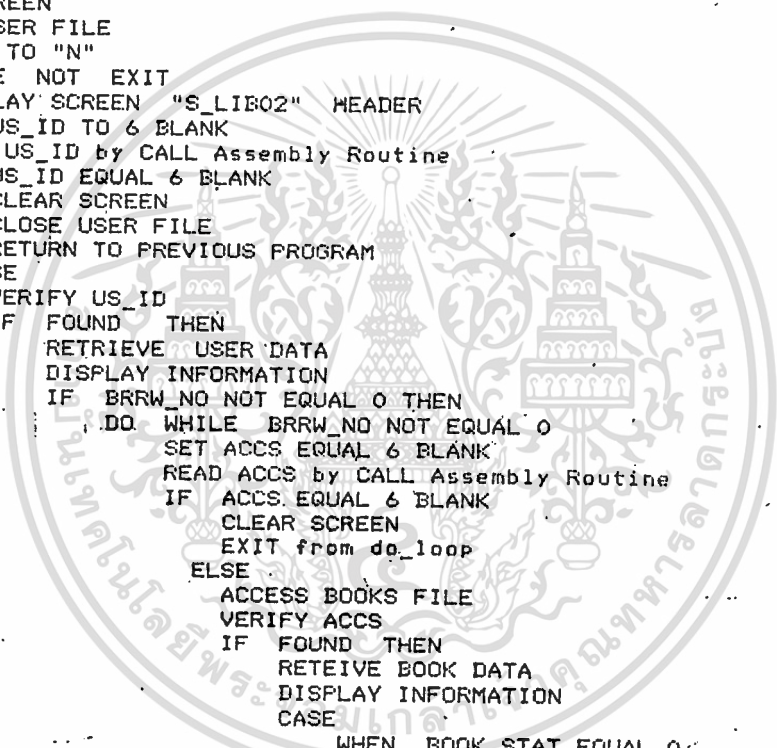
```
CLEAR SCREEN  
ACCESS BOOKS FILE  
SET EXIT TO "N"  
DO WHILE NOT EXIT  
  DISPLAY SCREEN "S_LIB01_4" HEADER  
  SET TITLE TO 6 BLANK  
  READ TITLE  
  IF TITLE EQUAL 6 BLANK  
    CLEAR SCREEN  
    CLOSE BOOKS FILE  
    RETURN TO PREVIOUS PROGRAM  
  ELSE  
    VERIFY TITLE  
    IF FOUND THEN  
      RETRIVE BOOK DATA  
      DISPLAY INFORMATION  
    ELSE  
      WRITE "THIS BOOK IS BORROWED by:"  
      ACCESS USER FILE  
      USE BW_ID TO VERIFY USER_ID  
      IF FOUND THEN  
        RETRIVE USER DATA  
        DISPLAY INFORMATION  
      ENDIF  
    ENDIF  
  ENDIF  
ENDDO
```

-----\*  
PROGRAM SPECIFICATION!  
-----\*

Task Name : BORROW THE BOOK  
Program Name : LIB02  
Program Description : Program borrow the book by specify USER\_ID & ACSS\_NO  
Input File : USER.DBF , BOOKS.DBF  
Output File : USER.DBF , BOOKS.DBF  
Screen : S J LIB02

[ Process Description ]

```
CLEAR SCREEN
ACCESS USER FILE
SET EXIT TO "N"
DO WHILE NOT EXIT
  DISPLAY SCREEN "S_LIB02" HEADER
  SET US_ID TO & BLANK
  READ US_ID by CALL Assembly Routine
  IF US_ID EQUAL & BLANK
    CLEAR SCREEN
    CLOSE USER FILE
    RETURN TO PREVIOUS PROGRAM
  ELSE
    VERIFY US_ID
    IF FOUND THEN
      RETRIEVE USER DATA
      DISPLAY INFORMATION
      IF BRRW_NO NOT EQUAL 0 THEN
        DO WHILE BRRW_NO NOT EQUAL 0
          SET ACSS EQUAL & BLANK
          READ ACSS by CALL Assembly Routine
          IF ACSS EQUAL & BLANK
            CLEAR SCREEN
            EXIT from do_loop
          ELSE
            ACCESS BOOKS FILE
            VERIFY ACSS
            IF FOUND THEN
              RETEIVE BOOK DATA
              DISPLAY INFORMATION
              CASE
                WHEN BOOK_STAT EQUAL 0
                  UPDATE DATA IN BOOKS FILE
                  UPDATE DATA IN USER FILE
                WHEN BOOK_STAT EQUAL 1
                  WRITE "THIS BOOK IS BORROWED"
                WHEN BOOK_STAT EQUAL 2
                  WRITE "THIS BOOK IS RESERVED"
              ELSE
                WRITE "THIS BOOK IS FUND ON FILE"
            ENDIF
          ENDIF
        ENDDO
      ELSE
        WRITE "YOU HAVE NO PERMITTED TO BORROW THE BOOK !"
      ENDIF
    ELSE
      WRITE "THIS USER IS NOT ON FILE OR LIBRARY"
    ENDIF
  ENDIF
ENDDO
```

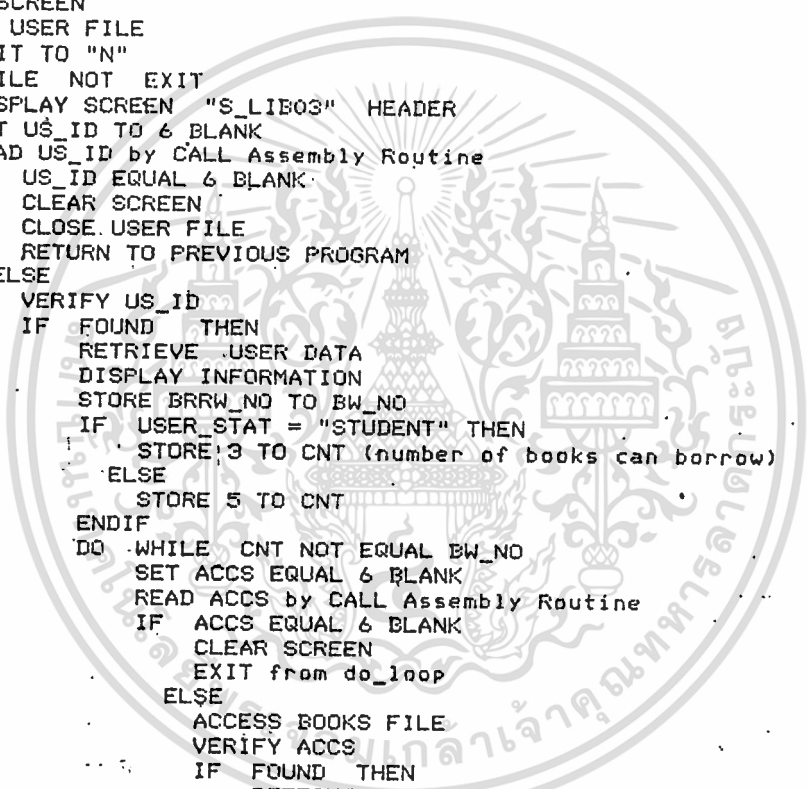


-----\*  
!PROGRAM SPECIFICATION!  
\*-----\*

Task Name : RETURN THE BOOK  
Program Name : LIB03  
Program Description : Program return the book by specify USER\_ID & ACSS\_NO  
Input File : USER.DBF , BOOKS.DBF  
Output File : USER.DBF , BOOKS.DBF  
Screen : S\_LIB03

[ Process Description ]

```
CLEAR SCREEN
ACCESS USER FILE
SET EXIT TO "N"
DO WHILE NOT EXIT
  DISPLAY SCREEN "S_LIB03" HEADER
  SET US_ID TO & BLANK
  READ US_ID by CALL Assembly Routine
  IF US_ID EQUAL & BLANK
    CLEAR SCREEN
    CLOSE USER FILE
    RETURN TO PREVIOUS PROGRAM
  ELSE
    VERIFY US_ID
    IF FOUND THEN
      RETRIEVE USER DATA
      DISPLAY INFORMATION
      STORE BRRW_NO TO BW_NO
      IF USER_STAT = "STUDENT" THEN
        STORE 3 TO CNT (number of books can borrow)
      ELSE
        STORE 5 TO CNT
      ENDIF
      DO WHILE CNT NOT EQUAL BW_NO
        SET ACSS EQUAL & BLANK
        READ ACSS by CALL Assembly Routine
        IF ACSS EQUAL & BLANK
          CLEAR SCREEN
          EXIT from do_loop
        ELSE
          ACCESS BOOKS FILE
          VERIFY ACSS
          IF FOUND THEN
            RETEIVE BOOK DATA
            DISPLAY INFORMATION
            ACCESS USER FILE
            UPDATE DATA IN USER FILE
            UPDATE DATA IN BOOKS FILE
          ELSE
            WRITE "THIS BOOK IS FUND ON FILE"
          ENDIF
        ENDIF
      ENDDO
    ELSE
      WRITE "THIS USER IS NOT ON FILE OR LIBRARY"
    ENDIF
  ENDIF
ENDDO
```

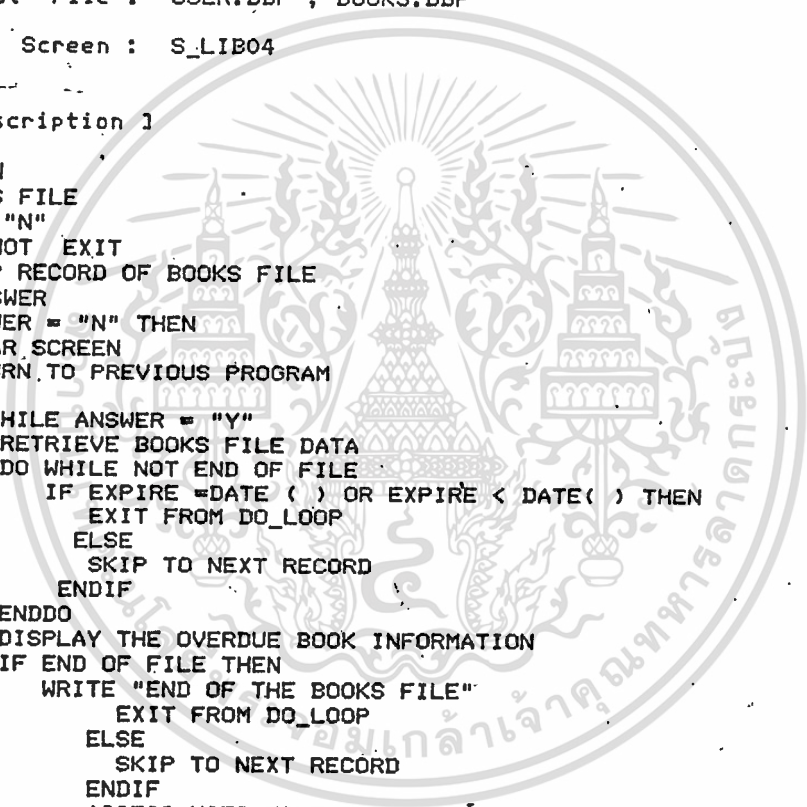


-----\*  
|PROGRAM SPECIFICATION|  
\*-----

Task Name : BORROWER DUE DATE  
Program Name : LIB04  
Program Description : Program to view the DUE DATE BOOKS  
Input File : USER.DBF , BOOKS.DBF  
Screen : S\_LIB04

[ Process Description ]

```
CLEAR SCREEN
ACCESS BOOKS FILE
SET EXIT TO "N"
DO WHILE NOT EXIT
GOTO TOP RECORD OF BOOKS FILE
READ ANSWER
IF ANSWER = "N" THEN
CLEAR SCREEN
RETURN TO PREVIOUS PROGRAM
ELSE
DO WHILE ANSWER = "Y"
RETRIEVE BOOKS FILE DATA
DO WHILE NOT END OF FILE
IF EXPIRE =DATE ( ) OR EXPIRE < DATE( ) THEN
EXIT FROM DO_LOOP
ELSE
SKIP TO NEXT RECORD
ENDIF
ENDDO
DISPLAY THE OVERDUE BOOK INFORMATION
IF END OF FILE THEN
WRITE "END OF THE BOOKS FILE"
EXIT FROM DO_LOOP
ELSE
SKIP TO NEXT RECORD
ENDIF
ACCESS USER FILE
USE BW_ID TO VERIFY USER_ID
IF FOUND THEN
RETRIVE USER DATA
DISPLAY INFORMATION
ENDIF
ENDDO
ENDIF
ENDDO
```



ภาคผนวกที่ ๕

แสดงโปรแกรมสำหรับใช้ในระบบห้องสมุด

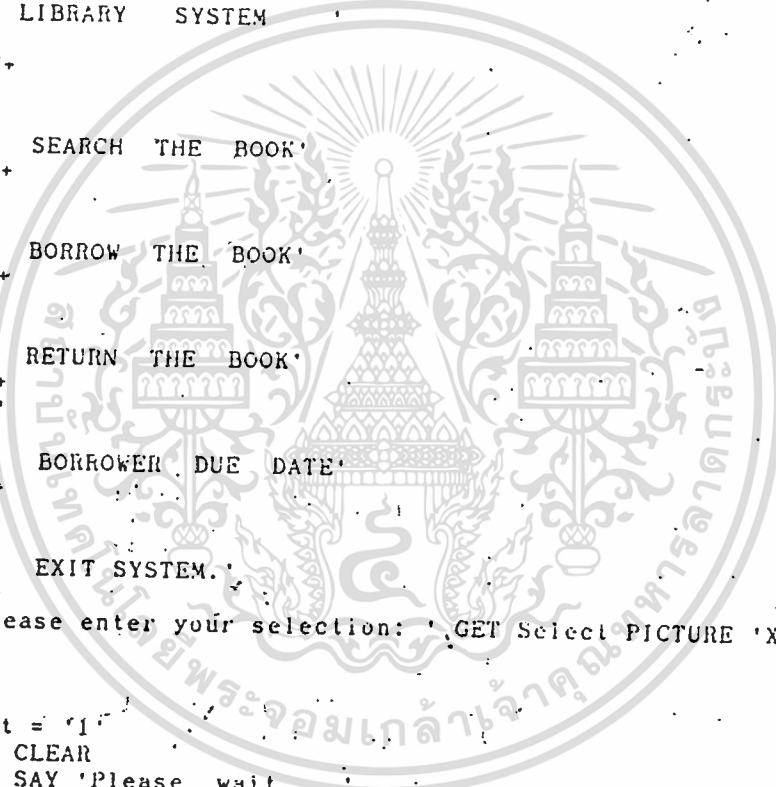


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

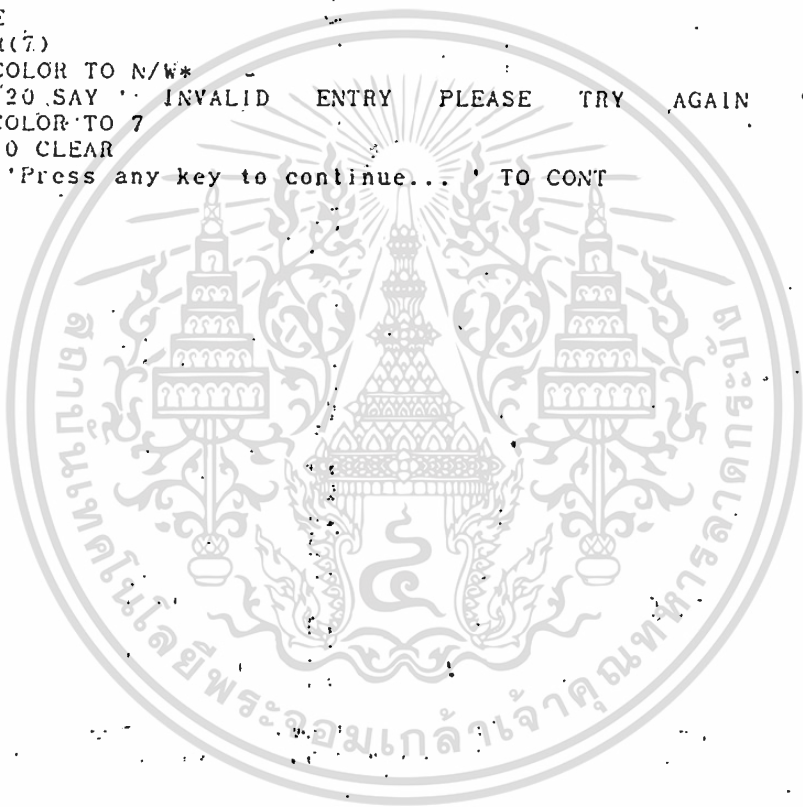
```

clear
@ 23,0 SAY 'Please wait...'
set talk off
load init8250
call init8250
release module init8250
DO WHILE .T.
  @ 21,0 CLEAR
  STORE ' ' TO Select
  SET COLOR TO W+
  @ 1,1 SAY ' DATE: '
  SET COLOR TO 7
  @ 1,10 SAY CDOW(DATE())+' '+CMONTH(DATE())+' '+LTRIM(STR(DAY(DATE()),2));
  + ' '+STR(YEAR(DATE()),4)
  SET COLOR TO G/W
  @ 1,57 SAY 'KMIT LADKRABANG LIBRARY'
  SET COLOR TO W+/W
  @ 4,26 SAY ' LIBRARY SYSTEM
  ?
  SET COLOR TO W+
  @ 7,15 SAY '1'
  SET COLOR TO 7
  @ 7,16 SAY ' SEARCH THE BOOK'
  SET COLOR TO W+
  @ 8,15 SAY '2'
  SET COLOR TO 7
  @ 8,16 SAY ' BORROW THE BOOK'
  SET COLOR TO W+
  @ 9,15 SAY '3'
  SET COLOR TO 7
  @ 9,16 SAY ' RETURN THE BOOK'
  SET COLOR TO W+
  @ 10,15 SAY '4'
  SET COLOR TO 7
  @ 10,16 SAY ' BORROWER DUE DATE'
  SET COLOR TO W+
  @ 14,15 SAY 'X'
  SET COLOR TO 7
  @ 14,16 SAY ' EXIT SYSTEM.'
  SET COLOR TO W+
  @ 17,40 SAY 'Please enter your selection: ' GET Select PICTURE 'X'
  READ
  SET COLOR TO 7
  DO CASE
    CASE Select = '1'
      @ 23,0 CLEAR
      @ 24,1 SAY 'Please wait...'
      DO SEARCH
    CASE Select = '2'
      @ 23,0 CLEAR
      @ 24,1 SAY 'Please wait...'
      DO BORROW
    CASE Select = '3'
      @ 23,0 CLEAR
      @ 24,1 SAY 'Please wait...'

```



```
DO RETURN
CASE Select = '4'
  @ 23,0 CLEAR
  @ 24,1 SAY 'Please wait... '
  DO DUEDATE
CASE UPPER(Select) = 'X'
  WAIT 'Are you sure ? (Y/N) : ' TO Select
  IF UPPER(Select) <> 'Y'
    @ 23,0 CLEAR
    @ 24,1 SAY 'Please wait... '
  ELSE
    @ 23,0 CLEAR
    @ 24,1 SAY 'Please, wait... '
    CLEAR
    QUIT
  ENDIF
OTHERWISE
  ? CHR(7)
  SET COLOR TO N/W*
  @ 21,20 SAY 'INVALID ENTRY PLEASE TRY AGAIN'
  SET COLOR TO 7
  @ 22,0 CLEAR
  WAIT 'Press any key to continue... ' TO CONT
ENDCASE
ENDDO
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLEAR
SET TALK OFF
SET DELIMITER OFF
DO WHILE .T.
  @ 21,0 CLEAR
  OPTION = ' '
  @ 1, 1 SAY 'DATE: '+DTC(DATE())
  SET COLOR TO N/W
  @ 1,57 SAY 'KMIT LADKRABANG LIBRARY'
  SET COLOR TO W+
  @ 2,32 SAY 'DDDDDDDDDDDDDDDDDD'
  @ 3,32 SAY ' SEARCH THE BOOK'
  @ 4,32 SAY 'DDDDDDDDDDDDDDDDDD'
  @ 7,15 SAY '1'
  SET COLOR TO 7
  @ 7,16 SAY ' . ACCESSION NUMBER'
  SET COLOR TO W+
  @ 8,15 SAY '2'
  SET COLOR TO 7
  @ 8,16 SAY ' . CALL NUMBER'
  SET COLOR TO W+
  @ 9,15 SAY '3'
  SET COLOR TO 7
  @ 9,16 SAY ' . AUTHOR'
  SET COLOR TO W+
  @ 10,15 SAY '4'
  SET COLOR TO 7
  @ 10,16 SAY ' . TITLE'
  SET COLOR TO W+
  @ 13,15 SAY 'E'
  SET COLOR TO 7
  @ 13,16 SAY ' . END'
  SET COLOR TO W+
  @ 19,40 SAY 'Please enter your selection: ' GET OPTION PICTURE 'X'
  READ
  DO CASE
    CASE OPTION = '1'
      @ 23,0 CLEAR
      @ 24,0 SAY 'Please wait...'
      DO ACCSSION
    CASE OPTION = '2'
      @ 23,0 CLEAR
      @ 24,0 SAY 'Please wait...'
      CLEAR
      DO CALL
    CASE OPTION = '3'
      @ 23,0 CLEAR
      @ 24,0 SAY 'Please wait...'
      CLEAR
      DO AUTHOR
    CASE OPTION = '4'
      @ 23,0 CLEAR
      @ 24,0 SAY 'Please wait...'
      CLEAR
      DO TITLE

```

```
CASE UPPER(OPTION) = 'E'  
  @ 23,0 CLEAR  
  @ 24,0 SAY 'Please wait...'  
  CLEAR  
  RETURN  
OTHERWISE  
  ? CHR(7)  
  SET COLOR TO N*/W  
  @ 21,20 SAY 'INVALID ENTRY PLEASE TRY AGAIN'  
  SET COLOR TO 7  
  @ 22,0 CLEAR  
  WAIT 'Press any key to continue... ' TO CONT  
ENDCASE  
ENDDO
```



```
CLEAR
LOAD receive
SET TALK OFF
SET DELIMITER OFF
STORE ' ' TO ACCS
select A
USE BOOKS INDEX ACCS_NO
select B
use USER index USER_ID
DO WHILE .T.
  select A
  @ 9,0 CLEAR
  SET COLOR TO 7
  @ 1, 1 SAY 'DATE: '+DTOC(DATE())
  SET COLOR TO N/W
  @ 1,57 SAY 'KMIT LADKRABANG LIBRARY'
  SET COLOR TO W+
  @ 3,32 SAY 'SEARCH THE BOOK'
  SET COLOR TO 7
  @ 5,0 SAY 'SEARCHING KEY IS -->'
  SET COLOR TO GR+
  @ 5,26 SAY 'ACCESSION NUMBER'
  SET COLOR TO 7
  ANS = ' '
  DO WHILE UPPER(ANS) # 'Y'
    ANS = ' '
    @ 8,27 CLEAR
    ACCS = ' '
    SET COLOR TO GR+
    @ 8,7 SAY 'ACCESSION NUMBER:'
    SET COLOR TO 7
    @ 23,0 CLEAR
    @ 23,0 SAY 'Press'
    SET COLOR TO GR+/W
    @ 23,7 SAY 'Esc'
    SET COLOR TO 7
    @ 23,13 SAY ' or '
    SET COLOR TO GR+/W
    @ 23,18 SAY 'Cr'
    SET COLOR TO 7
    @ 23,23 SAY 'to previous menu...'
    @ 8,27 SAY ''
    CALL receive WITH ACCS
    @ 8,27 SAY ACCS
    IF ACCS = ' '
      EXIT
    ENDIF
    @ 8,50 SAY 'That is OK? (Y/N): ' GET ANS PICTURE 'X'
  READ
ENDDO
IF ACCS = ' '
  @ 23,0 CLEAR
  @ 23,0 SAY 'Please wait...'
  USE
  RELEASE MODULE receive
```

```

CLEAR
RETURN
ELSE
@ 23,0 CLEAR
@ 23,0 SAY 'Pease wait...'
FIND &ACCS
IF .NOT. EOF()
@ 9,0 CLEAR
SET COLOR TO W+
@ 9,12 SAY 'CALL NUMBER:'
SET COLOR TO 7
@ 9,27 SAY CALL_NO
SET COLOR TO W+
@ 10,18 SAY 'AUTHOR:'
SET COLOR TO 7
@ 10,27 SAY AUTHOR
SET COLOR TO W+
@ 11,19 SAY 'TITLE:'
SET COLOR TO 7
@ 11,27 SAY LEFT(TITLE,5)
@ 11,19 SAY ' ' + RIGHT(TITLE,50)
SET COLOR TO W+
DO CASE
CASE BOOK_STAT = 0
@ 21,30 SAY ' THIS BOOK IS IN LIBRARY'
CASE BOOK_STAT = 1
@ 14,5 SAY ' THIS BOOK IS BORROWED by:'
bw_id = brrw_id
select B
find &bw_id
if .not. eof()
set color to w+
@ 16,22 say 'ID:'
@ 17,18 say 'STATUS:'
@ 18,20 say 'NAME:'
@ 19,15 say 'USER_DEPT:'
set color to w
@ 16,27 say bw_id
@ 17,27 say user_stat
@ 18,27 say user_name
@ 19,27 say user_dept
endif
CASE BOOK_STAT = 2
@ 21,30 SAY ' THIS BOOK IS RESERVED'
ENDCASE
SET COLOR TO 7
@ 22,0 CLEAR
WAIT 'Press any key to continue...' TO CONT
ELSE
? CHR(7)
@ 9,0 CLEAR
SET COLOR TO N*/W
@ 21,20 SAY ' THIS BOOK IS NOT ON FILE or LIBRARY'
SET COLOR TO 7
@ 22,0 CLEAR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WAIT 'Press any key to continue...' TO CONT
ENDIF
ENDIF
NDDO

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CLEAR
SET TALK OFF
SET DELIMITER OFF
STORE ' ' TO CALL_NUM
select A
USE BOOKS INDEX CALL
select B
use user index user_id
DO WHILE .T.
  select A
  @ 9,0 CLEAR
  SET COLOR TO 7
  @ 1, 1 SAY 'DATE: '+DTC(DATE())
  SET COLOR TO N/W
  @ 1,57 SAY 'KMIT LADKRABANG LIBRARY'
  SET COLOR TO W+
  @ 3,32 SAY 'SEARCH THE BOOK'
  SET COLOR TO 7
  @ 5,0 SAY 'SEARCHING KEY IS -->'
  SET COLOR TO GR+
  @ 5,26 SAY 'CALL NUMBER'
  SET COLOR TO 7
  ANS = ' '
  DO WHILE UPPER(ANS) # 'Y'
    ANS = ' '
    @ 8,0 CLEAR
    CALL_NUM = ' '
    SET COLOR TO GR+
    @ 9,12 SAY 'CALL NUMBER:'
    SET COLOR TO 7
    @ 9,27 GET CALL_NUM PICTURE 'XXXXXXXXXXXXXXXXX'
    @ 23,0 CLEAR
    @ 23,0 SAY 'Press'
    SET COLOR TO GR+/W
    @ 23,7 SAY 'Esc'
    SET COLOR TO 7
    @ 23,13 SAY 'or'
    SET COLOR TO GR+/W
    @ 23,18 SAY 'Cr'
    SET COLOR TO 7
    @ 23,23 SAY 'to previous menu...'
    READ
    IF CALL_NUM = ' '
      EXIT
    ENDIF
    @ 10,50 SAY 'That is OK ? (Y/N): ' GET ANS PICTURE 'X'
    READ
  ENDDO
  IF CALL_NUM = ' '
    @ 23,0 CLEAR
    @ 23,0 SAY 'Please wait...'
    USE
    CLEAR
    RETURN
  ELSE
```

```
@ 23,0 CLEAR
@ 23,0 SAY 'Peasse wait...'
FIND &CALL_NUM
IF .NOT. EOF()
  @ 9,0 CLEAR
  SET COLOR TO W+
  @ 8,7 SAY 'ACCESSION NUMBER:'
  SET COLOR TO 7
  @ 8,27 SAY ACCS_NO
  SET COLOR TO W+
  @ 9,12 SAY 'CALL NUMBER:'
  SET COLOR TO 7
  @ 9,27 SAY CALL_NO
  SET COLOR TO W+
  @ 10,18 SAY 'AUTHOR:'
  SET COLOR TO 7
  @ 10,27 SAY AUTHOR
  SET COLOR TO W+
  @ 11,19 SAY 'TITLE:'
  SET COLOR TO 7
  @ 11,27 SAY LEFT(TITLE,50)
  @ 12,19 SAY ' ' +RIGHT(TITLE,50)
  SET COLOR TO GR+*
  DO CASE
    CASE BOOK_STAT = 0
      @ 21,30 SAY ' THIS BOOK IS IN LIBRARY '
    CASE BOOK_STAT = 1
      @ 14,5 SAY ' THIS BOOK IS BORROWED by:'
      bw_id = brrw_id
      select B
      find &bw_id
      if .not. eof()
        set color to w+
        @ 16,22 say 'ID:'
        @ 17,18 say 'STATUS:'
        @ 18,20 say 'NAME:'
        @ 19,15 say 'USER_DEPT:'
        set color to w
        @ 16,27 say bw_id
        @ 17,27 say user_stat
        @ 18,27 say user_name
        @ 19,27 say user_dept
      endif
  ENDCASE
  SET COLOR TO 7
  @ 22,0 CLEAR
  WAIT 'Press any key to continue...' TO CONT
ELSE
  ? CHR(7)
  @ 9,0 CLEAR
  SET COLOR TO N*/W
  @ 21,20 SAY ' THIS BOOK IS NOT ON FILE or LIBRARY '
  SET COLOR TO 7
  @ 22,0 CLEAR
  WAIT 'Press any key to continue...' TO CONT
```

ENDIF  
ENDIF  
ENDDO



```
CLEAR
SET TALK OFF
SET DELIMITER OFF
AUTHR = '
select A ;
USE BOOKS INDEX AUTHOR
select B
use user index user_id
DO WHILE .T.
  select A
  @ 9,0 CLEAR
  SET COLOR TO 7
  @ 1, 1 SAY 'DATE: '+DTOC(DATE())
  SET COLOR TO N/W
  @ 1,57 SAY 'KMIT LADKRABANG LIBRARY'
  SET COLOR TO W+
  @ 3,32 SAY 'SEARCH THE BOOK'
  SET COLOR TO 7
  @ 5,0 SAY 'SEARCHING KEY IS -->'
  SET COLOR TO GR+
  @ 5,26 SAY 'AUTHOR'
  SET COLOR TO 7
  ANS = '
  DO WHILE UPPER(ANS) # 'Y'
    ANS = '
    @ 8,0 CLEAR
    AUTHR = '
    SET COLOR TO GR+
    @ 10,18 SAY 'AUTHOR:'
    SET COLOR TO 7
    @ 10,27 GET AUTHR
    @ 23,0 CLEAR
    @ 23,0 SAY 'Press'
    SET COLOR TO GR+/W
    @ 23,7 SAY 'Esc'
    SET COLOR TO 7
    @ 23,13 SAY 'or'
    SET COLOR TO GR+/W
    @ 23,18 SAY 'Cr'
    SET COLOR TO 7
    @ 23,23 SAY 'to previous menu...'
    READ
    IF AUTHR = '
      EXIT
    ENDIF
    @ 14,50 SAY 'That is OK? (Y/N): ' GET ANS PICTURE 'X'
    READ
  ENDDO
  IF AUTHR = '
    @ 23,0 CLEAR
    @ 23,0 SAY 'Please wait...'
    USE
    CLEAR
    RETURN
  ELSE
```

```
@ 23,0 CLEAR
@ 23,0 SAY 'Pease wait...'
FIND &AUTHR
IF .NOT. EOF()
  @ 9,0 CLEAR
  SET COLOR TO W+
  @ 8,7 SAY 'ACCESSION NUMBER:'
  SET COLOR TO 7
  @ 8,27 SAY ACCS_NO
  SET COLOR TO W+
  @ 9,12 SAY 'CALL NUMBER:'
  SET COLOR TO 7
  @ 9,27 SAY CALL_NO
  SET COLOR TO W+
  @ 10,18 SAY 'AUTHOR:'
  SET COLOR TO 7
  @ 10,27 SAY AUTHOR
  SET COLOR TO W+
  @ 11,19 SAY 'TITLE:'
  SET COLOR TO 7
  @ 11,27 SAY LEFT(TITLE,50)
  @ 12,19 SAY ' ' + RIGHT(TITLE,50)
  SET COLOR TO GR+*
  DO CASE
    CASE BOOK_STAT = 0
      @ 21,30 SAY ' THIS BOOK IS IN LIBRARY '
    CASE BOOK_STAT = 1
      @ 14,5 SAY ' THIS BOOK IS BORROWED by:'
      bw_id = brrw_id
      select B
      find &bw_id
      if .not. eof()
        set color to w+
        @ 16,22 say 'ID:'
        @ 17,18 say 'STATUS:'
        @ 18,20 say 'NAME:'
        @ 19,15 say 'USER_DEPT:'
        set color to w
        @ 16,27 say bw_id
        @ 17,27 say user_stat
        @ 18,27 say user_name
        @ 19,27 say user_dept
      endif
  ENDCASE
  SET COLOR TO 7
  @ 22,0 CLEAR
  WAIT 'Press any key to continue...' TO CONT
ELSE
  ? CHR(7)
  @ 9,0 CLEAR
  SET COLOR TO N*/W
  @ 21,20 SAY ' THIS BOOK IS NOT ON FILE or LIBRARY '
  SET COLOR TO 7
  @ 22,0 CLEAR
  WAIT 'Press any key to continue...' TO CONT
```

ENDIF  
ENDIF  
ENDDO



```
CLEAR
SET TALK OFF
SET DELIMITER OFF
TITLE1 =
TITLE2 =
select A
USE BOOKS INDEX TITLE
select B
use user index user_id
DO WHILE .T.
  select A
  @ 9,0 CLEAR
  SET COLOR TO 7
  @ 1, 1 SAY 'DATE: '+DTCO(DATE())
  SET COLOR TO N/W
  @ 1,57 SAY 'KMIT LADKRABANG LIBRARY'
  SET COLOR TO W+
  @ 3,32 SAY 'SEARCH THE BOOK'
  SET COLOR TO 7
  @ 5,0 SAY 'SEARCHING KEY IS --->'
  SET COLOR TO GR+
  @ 5,26 SAY 'TITLE'
  SET COLOR TO 7
  ANS =
  DO WHILE UPPER(ANS) # 'Y'
    ANS =
    @ 8,0 CLEAR
    TITLE1 =
    TITLE2 =
    SET COLOR TO GR+
    @ 11,19 SAY 'TITLE:'
    SET COLOR TO 7
    @ 11,27 GET TITLE1
    @ 12,27 GET TITLE2
    @ 23,0 CLEAR
    @ 23,0 SAY 'Press
    SET COLOR TO GR+/W
    @ 23,7 SAY 'Esc'
    SET COLOR TO 7
    @ 23,13 SAY ' or
    SET COLOR TO GR+/W
    @ 23,18 SAY ' Cr
    SET COLOR TO 7
    @ 23,23 SAY 'to previous menu...'
  READ
  IF TITLE1 =
    EXIT
  ENDIF
  @ 17,50 SAY 'That is OK ? (Y/N): ' GET ANS PICTURE 'X'
  READ
ENDDO
IF TITLE1 =
  @ 23,0 CLEAR
  @ 23,0 SAY 'Please wait...'
  USE
```

```
CLEAR
RETURN
ELSE
@ 23,0 CLEAR
@ 23,0 SAY 'Pease wait...'
TITLE3 = TITLE1 + TITLE2
FIND &TITLES
IF .NOT. EOF()
@ 9,0 CLEAR
SET COLOR TO W+
@ 8,7 SAY 'ACCESSION NUMBER:'
SET COLOR TO 7
@ 8,27 SAY ACCS_NO
SET COLOR TO W+
@ 9,12 SAY 'CALL NUMBER:'
SET COLOR TO 7
@ 9,27 SAY CALL_NO
SET COLOR TO W+
@ 10,18 SAY 'AUTHOR:'
SET COLOR TO 7
@ 10,27 SAY AUTHOR
SET COLOR TO W+
@ 11,19 SAY 'TITLE:'
SET COLOR TO 7
@ 11,27 SAY LEFT(TITLE,50)
@ 12,19 SAY ' ' + RIGHT(TITLE,50)
SET COLOR TO GR+*
DO CASE
CASE BOOK_STAT = 0
@ 21,30 SAY ' THIS BOOK IS IN LIBRARY '
CASE BOOK_STAT = 1
@ 14,5 SAY ' THIS BOOK IS BORROWED by:'
bw_id = brrw_id
select B
find &bw_id
if .not. eof()
set color to w+
@ 16,22 say 'ID:'
@ 17,18 say 'STATUS:'
@ 18,20 say 'NAME:'
@ 19,15 say 'USER_DEPT:'
set color to w
@ 16,27 say bw_id
@ 17,27 say user_stat
@ 18,27 say user_name
@ 19,27 say user_dept
endif
ENDCASE
SET COLOR TO 7
@ 22,0 CLEAR
WAIT 'Press any key to continue...' TO CONT
ELSE
? CHR(7)
@ 9,0 CLEAR
SET COLOR TO N*/W
```

@ 21,20 SAY ' THIS BOOK IS NOT ON FILE or LIBRARY '  
SET COLOR TO 7  
@ 22,0 CLEAR  
WAIT 'Press any key to continue... ' TO CONT'

ENDIF

ENDIF

ENDDO



```
CLEAR
LOAD receive
SET TALK OFF
SET DELIMITER OFF
@ 1,1 SAY 'DATE: '+DTC(DATE())
SET COLOR TO N/W
@ 1,57 SAY 'KMIT LADKRABANG LIBRARY'
SET COLOR TO W+
@ 2,31 SAY 'DDDDDDDDDDDDDDDDDD'
@ 3,32 SAY 'BORROW THE BOOK'
@ 4,31 SAY 'DDDDDDDDDDDDDDDDDD'
SET COLOR TO GR+
@ 5,10 SAY 'BORROWER '
SET COLOR TO 7
DO WHILE .T.
  @ 8,0 CLEAR
  USE USER INDEX USER_ID
  ANS = ' '
  DO WHILE UPPER(ANS) = 'Y'
    ANS = ' '
    @ 7,27 CLEAR
    US_ID = ' '
    SET COLOR TO W+
    @ 7,23 SAY 'ID: '
    SET COLOR TO 7
    @ 23,0 SAY 'Press '
    SET COLOR TO GR+/W
    @ 23,7 SAY ' Esc '
    SET COLOR TO 7
    @ 23,13 SAY ' or '
    SET COLOR TO GR+/W
    @ 23,18 SAY ' Cr '
    SET COLOR TO 7
    @ 23,23 SAY 'to previous menu...'
    @ 7,27 SAY ' '
    CALL receive WITH US_ID
    @ 7,27 SAY US_ID
    IF US_ID = ' '
      EXIT
    ENDIF
    @ 7,50 SAY 'That is OK ? (Y/N): ' GET ANS PICTURE 'X'
  READ
ENDDO
IF US_ID = ' '
  @ 23,0 CLEAR
  @ 23,0 SAY 'Please wait...'
  USE
  RELEASE MODULE receive
  CLEAR
  RETURN
ELSE
  @ 23,0 CLEAR
  @ 23,0 SAY 'Please wait...'
  FIND &US_ID
  IF .NOT. EOF()
```

```
SET COLOR TO W+
@ 8,19 SAY 'STATUS: '
SET COLOR TO 7
@ 8,27 SAY USER_STAT
SET COLOR TO W+
@ 9,21 SAY 'NAME: '
SET COLOR TO 7
@ 9,27 SAY USER_NAME
SET COLOR TO W+
@ 10,16 SAY 'USER_DEPT: '
SET COLOR TO 7
@ 10,27 SAY USER_DEPT
@ 23,0 CLEAR
IF BRRW_NO # 0
  BW_NO = BRRW_NO
  CNT = 0
  SET COLOR TO GR+
  @ 14,7 SAY 'BOOK DETAIL '
  SET COLOR TO 7
  DO WHILE CNT # BW_NO
    @ 17,0 CLEAR
    ANS = ' '
    DO WHILE UPPER(ANS) # 'Y'
      ANS = ' '
      @ 16,27 CLEAR
      ACCS = ' '
      SET COLOR TO W+
      @ 16,8 SAY 'ACCESSION NUMBER: '
      SET COLOR TO 7
      @ 23,0 SAY 'Press '
      SET COLOR TO GR+/W
      @ 23,7 SAY 'Esc '
      SET COLOR TO 7
      @ 23,13 SAY 'or '
      SET COLOR TO GR+/W
      @ 23,18 SAY 'Cr '
      SET COLOR TO 7
      @ 23,23 SAY 'to previous menu...'
      @ 16,27 SAY ' '
      CALL receive WITH ACCS
      @ 16,27 SAY ACCS
    IF ACCS = ' '
      EXIT
    ENDIF
    @ 16,50 SAY 'That is OK ? (Y/N): ' GET ANS PICTURE 'X'
    READ
  ENDDO
  IF ACCS = ' '
    @ 23,0 CLEAR
    @ 23,0 SAY 'Please wait...'
    EXIT
  ELSE
    @ 23,0 CLEAR
    @ 23,0 SAY 'Please wait...'
    USE BOOKS INDEX ACCS_NO
```

```
FIND &ACCS
IF .NOT. EOF()
  SET COLOR TO W+
  @ 17,13 SAY 'CALL NUMBER: '
  SET COLOR TO 7
  @ 17,27 SAY CALL_NO
  SET COLOR TO W+
  @ 18,19 SAY 'AUTHOR: '
  SET COLOR TO 7
  @ 18,27 SAY AUTHOR
  SET COLOR TO W+
  @ 19,20 SAY 'TITLE: '
  SET COLOR TO 7
  @ 19,27 SAY LEFT(TITLE,50)
  @ 20,19 SAY ' ' +RIGHT(TITLE,50)
DO CASE
  CASE BOOK_STAT = 0
    DUEDATE = DATE() + 7
    REPLACE EXPIRE WITH DUEDATE
    REPLACE BRRW_ID WITH US_ID
    REPLACE BOOK_STAT WITH 1
    USE
    USE USER INDEX USER_ID
    @ 23,0 CLEAR
    @ 23,0 SAY 'Pease wait...'
    FIND &US_ID
    IF .NOT. EOF()
      REPLACE BRRW_NO WITH BRRW_NO - 1
      CNT = CNT + 1
      DO CASE
        CASE BRRW_ACCS1 = '
          REPLACE BRRW_ACCS1 WITH ACCS
          USE
          @ 21,0 CLEAR
          SET COLOR TO GR+
          @ 22,0 SAY 'BORROW THE BOOK completed...'
          SET COLOR TO 7
          WAIT 'Press any key to continue...' TO CONT
        CASE BRRW_ACCS2 = '
          REPLACE BRRW_ACCS2 WITH ACCS
          USE
          @ 21,0 CLEAR
          SET COLOR TO GR+
          @ 22,0 SAY 'BORROW THE BOOK completed...'
          SET COLOR TO 7
          WAIT 'Press any key to continue...' TO CONT
        CASE BRRW_ACCS3 = '
          REPLACE BRRW_ACCS3 WITH ACCS
          USE
          @ 21,0 CLEAR
          SET COLOR TO GR+
          @ 22,0 SAY 'BORROW THE BOOK completed...'
          SET COLOR TO 7
          WAIT 'Press any key to continue...' TO CONT
        CASE BRRW_ACCS4 = '
          REPLACE BRRW_ACCS4 WITH ACCS
          USE
          @ 21,0 CLEAR
          SET COLOR TO GR+
          @ 22,0 SAY 'BORROW THE BOOK completed...'
          SET COLOR TO 7
          WAIT 'Press any key to continue...' TO CONT
```

```

REPLACE BRRW_ACCS4 WITH ACCS
USE
@ 21,0 CLEAR
SET COLOR TO GR+
@ 22,0 SAY ' BORROW THE BOOK completed.
SET COLOR TO 7
WAIT 'Press any key to continue... ' TO CONT
CASE BRRW_ACCS5 = '
REPLACE BRRW_ACCS5 WITH ACCS
USE
@ 21,0 CLEAR
SET COLOR TO GR+
@ 22,0 SAY ' BORROW THE BOOK completed...
SET COLOR TO 7
WAIT 'Press any key to continue... ' TO CONT

```

ENDCASE

ENDIF

```

CASE BOOK_STAT = 1
? CHR(7)
@ 21,0 CLEAR
SET COLOR TO N/W*
@ 22,25 SAY ' THIS BOOK IS BORROWED '
SET COLOR TO 7
WAIT 'Press any key to continue... ' TO CONT

```

```

CASE BOOK_STAT = 2
? CHR(7)
@ 21,0 CLEAR
SET COLOR TO N/W*
@ 22,25 SAY ' THIS BOOK IS RESERVED '
SET COLOR TO 7
WAIT 'Press any key to continue... ' TO CONT

```

ENDCASE

ELSE

```

? CHR(7)
@ 17,0 CLEAR
SET COLOR TO N/W*
@ 21,23 SAY ' THIS BOOK IS NOT FOUND ON FILE '
SET COLOR TO 7
@ 22,0 CLEAR
WAIT 'Press any key to continue... ' TO CONT

```

ENDIF

ENDIF

ENDDO

ELSE

```

? CHR(7)
@ 11,0 CLEAR
SET COLOR TO N/W*
@ 21,18 SAY ' You have no permitted to BORROW the Book ! '
SET COLOR TO 7
@ 22,0 CLEAR
WAIT 'Press any key to continue... ' TO CONT

```

ENDIF

ELSE

```

? CHR(7)
@ 21,0 CLEAR

```

```
SET COLOR TO N/W*  
Q 21,25 SAY ' THIS USER_ID NOT FOUND ON FILE  
SET COLOR TO 7  
@ 22,0 CLEAR  
WAIT 'Press any key to continue... ' TO CONT  
ENDIF  
ENDIF  
ENDDG
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLEAR
LOAD receive
SET TALK OFF
SET DELIMITER OFF
@ 1, 1 SAY 'DATE: '+DTOC( DATE() )
SET COLOR TO N/W
@ 1,57 SAY 'KNIT LADKRABANG LIBRARY'
SET COLOR TO W+
@ 2,31 SAY 'DDDDDDDDDDDDDDDDDD'
SET COLOR TO GR+
@ 3,32 SAY 'RETURN THE BOOK'
SET COLOR TO W+
@ 4,31 SAY 'DDDDDDDDDDDDDDDDDD'
SET COLOR TO W+
@ 5,10 SAY 'BORROWER '
SET COLOR TO 7
DO WHILE .T.
  USE USER INDEX USER_ID
  @ 8,0 CLEAR
  ANS = 'N'
  DO WHILE ANS = 'N'
    @ 7,27 CLEAR
    US_ID = ' '
    SET COLOR TO GR+
    @ 7,23 SAY 'ID: '
    SET COLOR TO 7
    @ 23,0 SAY 'Press '
    SET COLOR TO GR+/W
    @ 23,7 SAY 'Esc '
    SET COLOR TO 7
    @ 23,13 SAY ' or '
    SET COLOR TO GR+/W
    @ 23,18 SAY ' Cr '
    SET COLOR TO 7
    @ 23,23 SAY 'to previous menu...'
    @ 7,27 SAY ' '
    CALL receive WITH US_ID
    @ 7,27 SAY US_ID
    IF US_ID = ' '
      EXIT
    ENDIF
    @ 7,50 SAY 'That is OK ? (Y/N):' GET ANS PICTURE 'X'
  ...READ
  ENDDO
  IF US_ID = ' '
    @ 23,0 CLEAR
    @ 23,0 SAY 'Please wait...'
    USE
    RELEASE MODULE receive
    CLEAR
    RETURN
  ELSE
    @ 23,0 CLEAR
    @ 23,0 SAY 'Please wait...'
    FIND &US_ID
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน/เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF .NOT. EOF()
  SET COLOR TO GR+
  @ 8,19 SAY 'STATUS: '
  SET COLOR TO 7
  @ 8,27 SAY USER_STAT
  SET COLOR TO GR+
  @ 9,21 SAY 'NAME: '
  SET COLOR TO 7
  @ 9,27 SAY USER_NAME
  SET COLOR TO GR+
  @ 10,16 SAY 'USER_DEPT: '
  SET COLOR TO 7
  @ 10,27 SAY USER_DEPT
  @ 23,0 CLEAR
  BW_NO = BRRW_NO
  IF USER_STAT = 'STUDENT'
    CNT = 3
  ELSE
    CNT = 5
  ENDIF
  SET COLOR TO W+
  @ 14,7 SAY ' BOOK DETAIL '
  SET COLOR TO 7
  DO WHILE CNT # BW_NO
    @ 17,0 CLEAR
    ANS = 'N'
    DO WHILE ANS = 'N'
      @ 16,27 CLEAR
      ACCS = ' '
      SET COLOR TO GR+
      @ 16,11 SAY 'ACCESS NUMBER: '
      SET COLOR TO 7
      @ 23,0 SAY 'Press '
      SET COLOR TO GR+/W
      @ 23,7 SAY 'Esc '
      SET COLOR TO 7
      @ 23,13 SAY ' or '
      SET COLOR TO GR+/W
      @ 23,18 SAY 'Cr '
      SET COLOR TO 7
      @ 23,23 SAY 'to previous menu...'
      @ 16,27 SAY ''
      CALL receive WITH ACCS
      @ 16,27 SAY ACCS
      IF ACCS = ' '
        EXIT
      ENDIF
      @ 16,50 SAY 'That is OK ? (Y/N): ' GET ANS PICTURE 'X'
      READ
    ENDDO
    IF ACCS = ' '
      @ 23,0 CLEAR
      @ 23,0 SAY 'Please wait...'
      CLOSE DATABASES
      EXIT
    
```

```
ELSE
@ 23,0 CLEAR
@ 23,0 SAY 'Please wait...'
CLOSE DATABASES
USE BOOKS INDEX ACCS_NO
FIND &ACCS
IF .NOT. EOF()
  SET COLOR TO GR+
  @ 17,13 SAY 'CALL NUMBER: '
  SET COLOR TO 7
  @ 17,27 SAY CALL_NO
  SET COLOR TO GR+
  @ 18,19 SAY 'AUTHOR: '
  SET COLOR TO 7
  @ 18,27 SAY AUTHOR
  SET COLOR TO GR+
  @ 19,20 SAY 'TITLE: '
  SET COLOR TO 7
  @ 19,27 SAY LEFT(TITLE,50)
  @ 20,19 SAY ' ' + RIGHT(TITLE,50)
  CLOSE DATABASES
  USE USER INDEX USER_ID
  @ 23,0 CLEAR
  @ 23,0 SAY 'Please wait...'
  FIND &US_ID
  IF .NOT. EOF()
    REPLACE BRRW_NO WITH BRRW_NO + 1
    CNT = CNT - 1
    DO CASE
      CASE BRRW_ACCSI = ACCS
        REPLACE BRRW_ACCSI WITH '
        CLOSE DATABASES
        USE BOOKS INDEX ACCS_NO
        FIND &ACCS
        IF .NOT. EOF()
          EXPDATE = CTOD(' / / ')
          EXPDATE = EXPIRE
          IF BOOK_STAT = 1
            REPLACE EXPIRE WITH CTOD(' / / ')
            REPLACE BOOK_STAT WITH 0
            REPLACE BRRW_ID WITH '
          ELSE
            SET COLOR TO GR+*
            @ 21,25 SAY ' THIS BOOK WAS RESERVED
            SET COLOR TO 7
            REPLACE EXPIRE WITH CTOD(' / / ')
            REPLACE BRRW_ID WITH '
          ENDIF
        ENDIF
      ? CHR(7)
    @ 21,0 CLEAR
    IF EXPDATE < DATE()
      SET COLOR TO W+
      @ 21,25 SAY 'This book is overdue '
      SET COLOR TO GR+/W*
```

```
@ 21,48 SAY LTRIM(STR(DATE()-EXPDATE))
SET COLOR TO W+
@ 21,53 SAY 'days '
SET COLOR TO 7
ELSE
  IF EXPDATE = DATE()
    SET COLOR TO GR+/W
    @ 21,25 SAY ' This book is overdue today
    SET COLOR TO 7
  ENDIF
ENDIF
CLOSE DATABASES
SET COLOR TO W+
@ 22,0 SAY ' RETURN THE BOOK completed... '
SET COLOR TO 7
WAIT 'Press any key to continue... ' TO CONT
CASE BRRW_ACCS2 = ACCS
  REPLACE BRRW_ACCS2 WITH '
  CLOSE DATABASES
  USE BOOKS INDEX ACCS_NO
  FIND &ACCS
  IF .NOT. EOF()
    EXPDATE = CTOD(' / / ')
    EXPDATE = EXPIRE
    IF BOOK_STAT = 1
      REPLACE EXPIRE WITH CTOD(' / / ')
      REPLACE BOOK_STAT WITH 0
      REPLACE BRRW_ID WITH '
    ELSE
      SET COLOR TO GR+*
      @ 21,25 SAY ' THIS BOOK WAS RESERVED
      SET COLOR TO 7
      REPLACE EXPIRE WITH CTOD(' / / ')
      REPLACE BOOK_STAT WITH 0
      REPLACE BRRW_ID WITH '
    ENDIF
  ENDIF
  ? CHR(7)
  @ 21,0 CLEAR
  IF EXPDATE < DATE()
    SET COLOR TO W+
    @ 21,25 SAY 'This book is overdue '
    SET COLOR TO GR+/W*
    @ 21,48 SAY LTRIM(STR(DATE()-EXPDATE))
    SET COLOR TO W+
    @ 21,53 SAY 'days '
    SET COLOR TO 7
  ELSE
    IF EXPDATE = DATE()
      SET COLOR TO GR+/W
      @ 21,25 SAY ' This book is overdue today
      SET COLOR TO 7
    ENDIF
  ENDIF
ENDIF
CLOSE DATABASES
```

```
SET COLOR TO W+
@ 22,0 SAY ' RETURN THE BOOK completed...
SET COLOR TO 7
WAIT 'Press any key to continue... ' TO CONT
CASE BRRW_ACCS3 = ACCS
REPLACE BRRW_ACCS3 WITH '
CLOSE DATABASES
USE BOOKS INDEX ACCS_NO
FIND &ACCS
IF .NOT. EOF()
EXPDATE = CTOD(' / / ')
EXPDATE = EXPIRE
IF BOOK_STAT = 1
REPLACE EXPIRE WITH CTOD(' / / ')
REPLACE BOOK_STAT WITH 0
REPLACE BRRW_ID WITH '
ELSE
SET COLOR TO GR+*
@ 21,25 SAY ' THIS BOOK WAS RESERVED
SET COLOR TO 7
REPLACE EXPIRE WITH CTOD(' / / ')
REPLACE BOOK_STAT WITH 0
REPLACE BRRW_ID WITH '
ENDIF
ENDIF
? CHR(7)
@ 21,0 CLEAR
IF EXPDATE < DATE()
SET COLOR TO W+
@ 21,25 SAY 'This book is overdue
SET COLOR TO GR+/*
@ 21,48 SAY LTRIM(STR(DATE()-EXPDATE))
SET COLOR TO W+
@ 21,53 SAY 'days
SET COLOR TO 7
ELSE
IF EXPDATE = DATE()
SET COLOR TO GR+/*
@ 21,25 SAY ' This book is overdue today
SET COLOR TO 7
ENDIF
ENDIF
CLOSE DATABASES
SET COLOR TO W+
@ 22,0 SAY ' RETURN THE BOOK completed...
SET COLOR TO 7
WAIT 'Press any key to continue... ' TO CONT
CASE BRRW_ACCS4 = ACCS
REPLACE BRRW_ACCS4 WITH '
CLOSE DATABASES
USE BOOKS INDEX ACCS_NO
FIND &ACCS
IF .NOT. EOF()
EXPDATE = CTOD(' / / ')
EXPDATE = EXPIRE
```

```
IF BOOK_STAT = 1
  REPLACE EXPIRE WITH CTOD(' / / ')
  REPLACE BOOK_STAT WITH 0
  REPLACE BRRW_ID WITH '
ELSE
  SET COLOR TO GR+*
  @ 21,25 SAY ' THIS BOOK WAS RESERVED
  SET COLOR TO 7
  REPLACE EXPIRE WITH CTOD(' / / ')
  REPLACE BOOK_STAT WITH 0
  REPLACE BRRW_ID WITH '
ENDIF
ENDIF
? CHR(7)
@ 21,0 CLEAR
IF EXPDATE < DATE()
  SET COLOR TO W+
  @ 21,25 SAY 'This book is overdue
  SET COLOR TO GR+/W*
  @ 21,18 SAY LTRIM(STR(DATE()-EXPDATE))
  SET COLOR TO W+
  @ 21,53 SAY 'days '
  SET COLOR TO 7
ELSE
  IF EXPDATE = DATE()
    SET COLOR TO GR+/W
    @ 21,25 SAY ' This book is overdue today
    SET COLOR TO 7
  ENDIF
ENDIF
CLOSE DATABASE
SET COLOR TO W+
@ 22,0 SAY ' RETURN THE BOOK completed...
SET COLOR TO 7
WAIT 'Press any key to continue... ' TO CONT
CASE
  BRRW_ACCS5 = ACCS
  REPLACE BRRW_ACCS5 WITH
  CLOSE DATABASES
  USE BOOKS INDEX ACCS_NO
  FIND &ACCS
  IF .NOT. EOF()
    EXPDATE = CTOD(' / / ')
    EXPDATE = EXPIRE
    IF BOOK_STAT = 1
      REPLACE EXPIRE WITH CTOD(' / / ')
      REPLACE BOOK_STAT WITH 0
      REPLACE BRRW_ID WITH '
    ELSE
      SET COLOR TO GR+*
      @ 21,25 SAY ' THIS BOOK WAS RESERVED
      SET COLOR TO 7
      REPLACE EXPIRE WITH CTOD(' / / ')
      REPLACE BOOK_STAT WITH 0
      REPLACE BRRW_ID WITH '
    ENDIF
  ENDIF
```



```
CLEAR
SET TALK OFF
SET DELIMITER OFF
@ 1, 1 SAY 'DATE: '+DTC(DATE())
SET COLOR TO N/W
@ 1, 57 SAY 'KMIT LADKRABANG LIBRARY'
SET COLOR TO W+
@ 2, 31 SAY 'DDDDDDDDDDDDDDDDDDDDDDDDDDDD!'
SET COLOR TO GR+
@ 3, 31 SAY 'BORROWER DUE DATE'
SET COLOR TO W+
@ 4, 31 SAY 'DDDDDDDDDDDDDDDDDDDDDDDDDDDD'
SELECT A
USE BOOKS
DO WHILE .T.
  @ 5, 0 CLEAR
  GO TOP
  ANS = ''
  DO WHILE UPPER(ANS) # 'Y'
    SET COLOR TO 7
    ANS = ''
    @ 5, 74 CLEAR
    @ 5, 5 SAY 'Would you like to view The DUE DATE BOOKS for today ... (Y/N):'
    @ 5, 74 GET ANS.PICTURE 'X'
    SET COLOR TO 7
    @ 23, 0 SAY 'Press '
    SET COLOR TO GR+/W
    @ 23, 7 SAY 'Esc '
    SET COLOR TO 7
    @ 23, 13 SAY ' or '
    SET COLOR TO GR+/W
    @ 23, 18 SAY 'Cr '
    SET COLOR TO 7
    @ 23, 23 SAY 'to previous menu...'
    READ
    IF UPPER(ANS) = ' ' .OR. UPPER(ANS) = 'N'
      ANS = ''
      EXIT
    ENDIF
  ENDDO
  IF UPPER(ANS) = 'Y'
    @ 23, 0 CLEAR
    @ 23, 0 SAY 'Please wait...'
    SELECT A
    USE
    SELECT B
    USE
    CLEAR
    RETURN
  ELSE
    DO WHILE UPPER(ANS) = 'Y'
      SELECT A
      @ 6, 0 CLEAR
      @ 23, 0 SAY 'Please wait...'
      DO WHILE .NOT. EOF()
```

```
IF EXPIRE = DATE() .OR. EXPIRE < DATE()
  EXIT
  ELSE
  SKIP
ENDIF
ENDDO
SET COLOR TO GR+/W
@ 14,7 SAY ' BOOK DETAIL '
SET COLOR TO 7
SET COLOR TO W+
@ 16,7 SAY 'ACCESSION NUMBER: '
SET COLOR TO 7
@ 16,27 SAY ACCS_NO
SET COLOR TO W+
@ 17,12 SAY 'CALL NUMBER: '
SET COLOR TO 7
@ 17,27 SAY CALL_NO
SET COLOR TO W+
@ 18,18 SAY 'AUTHOR: '
SET COLOR TO 7
@ 18,27 SAY AUTHOR
SET COLOR TO W+
@ 19,19 SAY 'TITLE: '
SET COLOR TO 7
@ 19,27 SAY LEFT(TITLE,50)
@ 20,19 SAY ' ' '+RIGHT(TITLE,50)
IF EXPIRE < DATE()
  SET COLOR TO W+
  @ 21,25 SAY 'This book is overdue'
  SET COLOR TO GR+/W*
  @ 21,48 SAY LTRIM(STR(DATE()-EXPIRE))
  SET COLOR TO W+
  @ 21,53 SAY 'days'
  SET COLOR TO 7
ELSE
  SET COLOR TO GR+/W
  @ 21,25 SAY ' This book is overdue today'
  SET COLOR TO 7
ENDIF
BW_ID = BRRW_ID
IF EOF()
  @ 6,0 CLEAR
  SET COLOR TO N/W*
  @ 21,25 SAY ' END of the BOOKS file ....!'
  SET COLOR TO 7
  @ 22,0 CLEAR
  WAIT 'Press any key to continue ... ' TO CONT
  EXIT
ELSE
  SKIP
ENDIF
SELECT B
USE USER INDEX USER_ID
FIND &BW_ID
IF .NOT. EOF()
```

```
SET COLOR TO GR+/W
@ 6,10 SAY 'BORROWER '
SET COLOR TO W+
@ 7,22 SAY 'ID: '
SET COLOR TO 7
@ 7,27 SAY USER_ID
SET COLOR TO W+
@ 8,18 SAY 'STATUS: '
SET COLOR TO 7
@ 8,27 SAY USER_STAT
SET COLOR TO W+
@ 9,20 SAY 'NAME: '
SET COLOR TO 7
@ 9,27 SAY USER_NAME
SET COLOR TO W+
@ 10,15 SAY 'USER_DEPT: '
SET COLOR TO 7
@ 10,27 SAY USER_DEPT
ENDIF
@ 23,0 CLEAR
ANS = ' '
@ 23,0 SAY 'Would you like to view any more ? (Y/N): ' GET ANS PICTURE 'X'
READ
IF UPPER(ANS) = 'Y'
    SELECT A
    GO TOP
ENDIF
ENDDO
ENDIF
DDO
```

```
-----  
CODE      SEGMENT  PARA PUBLIC 'CODE'  
START     PROC      FAR  
;Standard program prologue  
;  
      ASSUME  CS:CODE  
;  
;Part1: Initialize the 8250 UART for :  
;      7 data bits, 1 stop bit, odd parity, and 1200 baud  
;      set up for loop back feature  
;  
      MOV     DX,3FBH      ; Address of line control register  
      MOV     AL,80H  
      OUT    DX,AL        ; To address baud rate divisor registers  
      MOV     DX,3F8H      ; Address of baud rate divisor LSB  
      MOV     AL,18H       ; LSB value for 4800 baud  
      OUT    DX,AL  
      MOV     DX,3F9H      ; Address of baud rate divisor MSB  
      MOV     AL,0         ; MSB value for 4800 baud  
      OUT    DX,AL  
; The baud rate has now been initialized  
; Now initialize the line control register  
      MOV     DX,3FBH      ; Address of line control register  
      MOV     AL,07H       ; NO parity, 2 stop bit, 8 data bits  
      OUT    DX,AL  
; Now initialize the MODEM control register for :  
;      request to SEND and data terminal READY signals  
;      and set MODEM control signals  
;  
      MOV     DX,3FCH      ; Address of MODEM control register  
      MOV     AL,03H       ; Set MODEM control signals  
      OUT    DX,AL  
; Now disable all four classes of interrupts  
      MOV     DX,3F9H      ; Address of INTERRUPT ENABLE register  
      MOV     AL,0  
      OUT    DX,AL  
; The initialization of the 8250 is now complete  
;  
      RET  
START     ENDP  
CODE      ENDS  
;      END      START  
-----
```

SEGMENT PARA PUBLIC 'CODE'  
PROC FAR

Standard program prologue

ASSUME CS:CODE

ts2: Received DATA from the 8250 for form keyboard  
then pass it to DBASEIII-plus program.

```

MOV     CX,6           ; Number of character to be passed to DBASEIII-plus PROGRAM
MOV     DX,3FDH        ; Address of line status register
IN      AL,DX          ; Line status register in AL
TEST    AL,1EH         ; Test for a reception error
JNZ     ERROR          ; Received error handling
TEST    AL,01H         ; Test for received data ready
JNZ     RECEIVE        ; Received character handling

```

FALL through to here the 8250 is ready for another  
character to get form keyboard  
the keyboard buffer has a character then get it and  
pass it to DBASEIII-plus PROGRAM

```

MOV     AH,1           ; Check keyboard buffer code
INT     16H            ; BIOS call
JZ      _MAIN          ; If no keyboard character nothing to pass

```

FALL through to here then there is a keyboard character to transmit

```

MOV     AH,0           ; Get keyboard input code
INT     16H            ; BIOS call
CMP     AL,1BH         ; was it an ESCAPE ?
JZ      _RET           ; Yes. go to DBASEIII-plus PROGRAM
CMP     AL,0DH         ; was it a carriage return ?
JZ      _RET           ; Yes. go to DBASEIII-plus PROGRAM
PUSH    BX
MOV     BX,0
MOV     AH,14
INT     10H
POP     BX
CMP     AL,08H         ; was it a back space ?
JZ      BS            ; Yes. go to get next character

```

has the keyboard character for pass to DBASEIII-plus PROGRAM

```

MOV     [BX],AL        ; Pass the character to DBASEIII-plus PROGRAM
INC     BX             ; Point to next character
LOOP   _MAIN           ; Do it 5 times

```

RET ; Return to DBASEIII-plus PROGRAM

DEC BX  
INC CX

JMP KEYBD

: This point is reached if a character was received by the 8250  
: with no error

RECEIVE: MOV DX,3F8H ; Address of the receiver data register  
IN AL,DX ; AL has the received character  
MOV [BX],AL ; Passs the character to DBASEIII-plus PROGRAM  
INC BX ; Point to next character  
JMP \_NEXT ; To receive next character

: If this point is reached then there was an error in the received  
: character. Clear the receiver data register and display a ?

ERROR: MOV DX,3F8H ; Address of the receiver data register  
IN AL,DX ; AL has the incorrect character  
MOV AL,'?' ; Display a '?' instead  
MOV [BX],AL  
INC BX  
JMP \_NEXT

:  
START ENDP  
CODE ENDS  
END START



ภาคผนวกที่ 6

วิธีใช้เครื่องอ่านบาร์โค้ด และไมโครคอมพิวเตอร์ในงานห้องสมุด



**TYPE TIL180  
BAR-CODE READ HEAD**

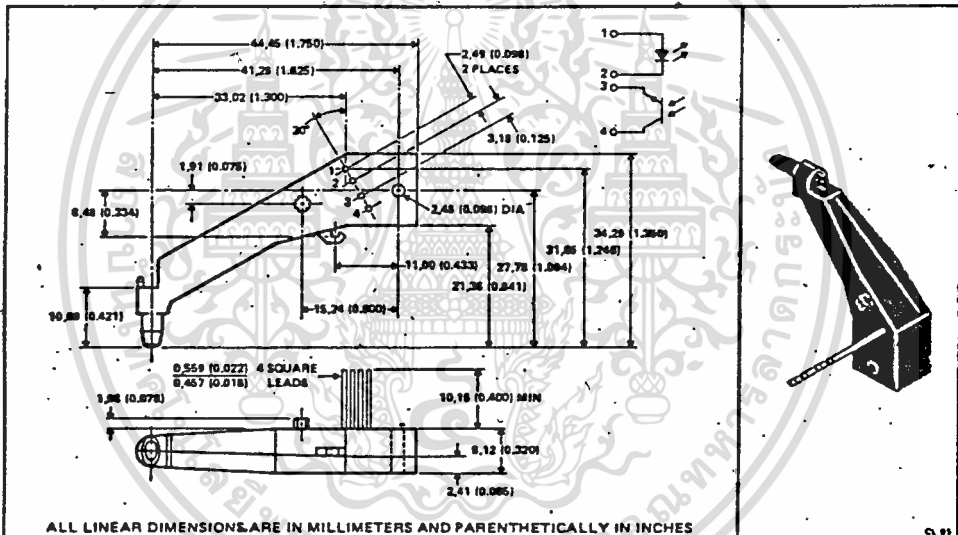
D2742, APRIL 1983

**INFRARED SENSOR AND EMITTER FOR BAR-CODE READING APPLICATIONS**

- Capable of Reading Black and White Bar Codes, i.e., UPC, EAN, Code 39, HP, and MSI
- Designed PCB for Mounting
- Contains a Gallium Arsenide Infrared LED and Phototransistor
- Reads Offset Press, Dot Matrix, and Printed Codes
- Codes Must Be Printed with Inks with a High Carbon Content

**mechanical data**

Each assembly contains a Gallium Arsenide Diode that emits light in the 940-nm region and a silicon phototransistor detector. The case is made of high-impact polycarbonate plastic. The assembly weight is approximately 5 grams.



**absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)**

Source Reverse Voltage	2 V
Source Continuous Forward Current	100 mA
Source Peak Forward Current (see Note 1)	1 A
Sensor Collector-Emitter Voltage	25 V
Sensor Emitter-Collector Voltage	5 V
Sensor Continuous Dissipation at (or below) 25°C Free-Air Temperature (see Note 2)	90 mW
Operating Free-Air Temperature Range	-40°C to 70°C
Storage Temperature Range	-40°C to 70°C
Lead Temperature 1,6 mm (1/16 inch) from Assembly for 5 Seconds	240°C

- NOTES: 1. This value applies for  $t_{pw} \leq 1 \mu s$ , PRR  $\leq 300$  pps.  
2. Derate linearly to 70°C free-air temperature at the rate of 2 mW/°C.

**TEXAS INSTRUMENTS**  
INCORPORATED

POST OFFICE BOX 225012 • DALLAS, TEXAS 75268

Copyright © 1983 by Texas Instruments Incorporated

8-27

**TYPE TIL180  
BAR-CODE READ HEAD**

electrical characteristics at 25°C free-air temperature

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
V(BR)CEO Collector-Emitter Breakdown Voltage	I <sub>C</sub> = 100 μA, I <sub>F</sub> = 0	25			V
V(BR)ECO Emitter-Collector Breakdown Voltage	I <sub>E</sub> = 100 μA, I <sub>F</sub> = 0	5			V
I <sub>C(off)</sub> Off-State Collector Current	V <sub>CE</sub> = 10 V, I <sub>F</sub> = 0			200	nA
I <sub>C(on)</sub> On-State Collector Current, (White Paper)	V <sub>CE</sub> = 5 V, I <sub>F</sub> = 40 mA, See Note 3	5	30		μA
V <sub>F</sub> Input Diode Static Forward Voltage	I <sub>p</sub> = 50 mA		1.2	1.45	V
η Reading Efficiency (see Note 4)	V <sub>CE</sub> = 5 V, I <sub>F</sub> = 40 mA	65%		100%	

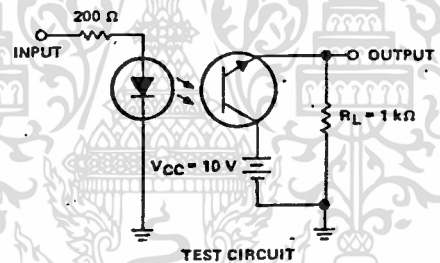
NOTES: 3. The reflective surface is 9-point chromate paper coated on both sides with low gloss varnish less than 0.00076 mm (0.0003 inch) thick.  
4. This is ratio of (1) the peak-to-peak change in collector current when the red head is scanning a test bar-code pattern to (2) the difference in I<sub>C(on)</sub> with the read head over white paper and over inked paper. The scanning rate is 767 mm/s (30 in/s), the bar code pattern is comprised of 0.254-mm (0.010-in) bars and spaces, and the ink is Pantone 419C or other high-carbon black ink.

switching characteristics at 25°C free-air temperature

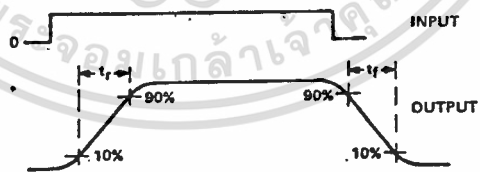
PARAMETER	TEST CONDITIONS <sup>1</sup>	MIN	TYP	MAX	UNIT
t <sub>r</sub> Rise Time	V <sub>CC</sub> = 10 V, I <sub>C(on)</sub> = 1 mA, R <sub>L</sub> = 1 kΩ, See Figure 1		125		μs
t <sub>f</sub> Fall Time			125		μs

<sup>1</sup>Stray irradiation outside the range of device sensitivity may be present. A satisfactory condition has been achieved when the parameter being measured approaches a value that cannot be altered by further irradiation shielding.

**PARAMETER MEASUREMENT INFORMATION**



ADJUST AMPLITUDE OF INPUT PULSE FOR I<sub>C(on)</sub> = 1 mA



NOTE: The input pulse is supplied by a generator having the following characteristics: Z<sub>out</sub> = 50 Ω, t<sub>r</sub> < 100 ns, t<sub>f</sub> < 100 ns, t<sub>w</sub> = 10 μs, duty cycle = 2%.

**VOLTAGE WAVEFORMS**  
**FIGURE 1—SWITCHING TIMES**

**SDS**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้